



Universidad
Carlos III de Madrid

INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

SOLUCIÓN DE ALMACENAMIENTO DISTRIBUIDO PARA SINCRONIZACIÓN DE FICHEROS EN DISPOSITIVOS ANDROID

Autor: Alberto Plaza López

Tutor: Francisco Javier García Blas

Leganés, Enero de 2014

RESUMEN

El gran auge con el que cuentan actualmente los dispositivos móviles, como *tablets* y *smartphones*, ha provocado que muchos usuarios incluso posean varios de estos dispositivos y que los usen en cualquier lugar en el que se encuentren.

Además, se han producido grandes avances en los sistemas operativos móviles, muchos de ellos de código abierto que permiten extender la funcionalidad a través de aplicaciones de terceros, siendo Android el más popular. También se han producido grandes avances en las características de los propios dispositivos móviles, que han aumentado su capacidad de almacenamiento y su potencia de procesamiento, así como en las redes de comunicaciones móviles, principalmente para la conexión a internet móvil, que ha posibilitado el uso de aplicaciones móviles que requieren conexión a internet desde cualquier lugar.

Todo ello ha hecho necesario para muchos usuarios el poder disponer de sus ficheros en todos sus dispositivos y en cualquier lugar que se encuentren. Para cubrir esta necesidad de los usuarios, en este documento se detalla el desarrollo de un sistema de almacenamiento, distribución y sincronización de ficheros que los usuarios pueden utilizar en dispositivos que funcionen con el sistema operativo Android. Este sistema permite tener los ficheros disponibles en todos los dispositivos que utilice el usuario y sincronizar de forma automática los cambios realizados en los ficheros en todos los dispositivos. Además, este sistema cuenta con la particularidad de que dispone de un sistema de etiquetas. Este sistema permite al usuario asignar etiquetas a los ficheros, de forma que se posibilita su clasificación y la navegación por los mismos. Esto simplifica la tarea de encontrar un determinado fichero o un conjunto de ellos.

Palabras clave: Android, ficheros, almacenamiento distribuido, sincronización de ficheros, distribución de ficheros, Dropbox, etiquetas.

ABSTRACT

The current boom of mobile devices, like smartphones and tablets, has caused many users owning several of them and using them anywhere they are.

Also, there have been great improvements in the mobile operating systems, many of them open source, which allow extending their functionality through third-party applications, being Android the most popular one. There have also been great improvements in the mobile devices characteristics, which have increased their storage capacity and processing power, and in the mobile communication networks, mainly the mobile internet connection, which have made possible using mobile applications which require internet connection anywhere.

All of that have made necessary to many users being able to have their files available in all their devices and anywhere they are. To meet this user need, this document details the development of a file storage, distribution and synchronization system which can be used by the users in devices running Android operating system. This system allows having the files available in all user devices and automatically synchronizes all changes made to files in all devices. Also, this system has the distinctive feature of having a label system. This system allows the user to assign labels to the files to make easier their classification and the navigation among them. This simplifies finding a specific file or a group of them.

Keywords: Android, files, distributed storage, file synchronization, file distribution, Dropbox, labels.

ÍNDICE DE CONTENIDOS

1.	INTRODUCCIÓN	12
1.1.	Motivación	12
1.2.	Objetivos	12
1.3.	Estructura de la memoria.....	13
1.4.	Glosario de términos.....	14
2.	ESTADO DEL ARTE	17
2.1.	Android.....	17
2.2.	Java.....	18
2.3.	Eclipse.....	19
2.4.	XML.....	20
2.5.	Análisis de soluciones en el mercado.....	22
2.5.1.	Dropbox.....	24
2.5.2.	Google Drive	26
2.5.3.	Ubuntu One Files.....	29
2.5.4.	Microsoft SkyDrive	31
2.5.5.	Resumen de características de las soluciones analizadas	33
2.5.6.	Conclusiones sobre las soluciones analizadas.....	35
3.	ANÁLISIS DEL SISTEMA	37
3.1.	Descripción general.....	37
3.1.1.	Navegación por etiquetas	38
3.2.	Casos de Uso	39
3.2.1.	Diagrama de Casos de Uso	39
3.2.2.	Especificación de Casos de Uso.....	40
3.3.	Requisitos del sistema.....	50
3.3.1.	Requisitos funcionales.....	52
3.3.2.	Requisitos no funcionales	56
3.3.3.	Requisitos de restricción	58
4.	DISEÑO DEL SISTEMA	59
4.1.	Arquitectura del sistema	59
4.1.1.	Arquitectura general	60
4.1.2.	Arquitectura del servidor	61
4.1.3.	Arquitectura del cliente.....	61

4.2.	Diagramas de clases	63
4.3.	Diagramas de secuencia	67
4.3.1.	Registrar usuario	68
4.3.2.	Iniciar sesión de usuario	69
4.3.3.	Importar fichero	70
4.3.4.	Abrir fichero	71
4.3.5.	Editar fichero	72
4.3.6.	Borrar fichero	73
4.3.7.	Modificar etiquetas	74
4.3.8.	Filtrar ficheros por etiqueta	75
5.	IMPLEMENTACIÓN DEL SISTEMA	76
5.1.	Protocolo de comunicación	76
5.1.1.	Especificación general de los mensajes	78
5.1.2.	Mensaje de Signup	79
5.1.3.	Mensaje de Accountdelete	80
5.1.4.	Mensaje de Login	81
5.1.5.	Mensaje de List	82
5.1.6.	Mensaje de Date	85
5.1.7.	Mensaje de Get	85
5.1.8.	Mensaje de Put	86
5.1.9.	Mensaje de Update	88
5.1.10.	Mensaje de Labelupdate	89
5.1.11.	Mensaje de Delete	91
5.2.	Ficheros	93
5.2.1.	Fichero de metadatos del servidor	94
5.2.2.	Fichero de metadatos del cliente	95
5.3.	Codificación de datos	99
5.4.	Sockets e hilos del sistema	102
5.4.1.	Sockets del sistema	102
5.4.2.	Hilos de la aplicación servidor	103
5.4.3.	Hilos de la aplicación cliente	103
5.5.	Otras consideraciones en la implementación	105
5.5.1.	Modelo de metadatos del sistema	105
5.5.2.	Sistema de control de versiones y conflictos de fichero	106

6.	PRUEBAS.....	107
6.1.	Dispositivos de prueba	107
6.2.	Pruebas de aceptación	109
6.3.	Matriz de trazabilidad Requisitos Funcionales – Pruebas.....	115
7.	PLANIFICACIÓN Y PRESUPUESTO	116
7.1.	Planificación	116
7.2.	Presupuesto	119
7.2.1.	Costes de personal	119
7.2.2.	Costes de hardware.....	121
7.2.3.	Costes de software	122
7.2.4.	Coste total del proyecto.....	123
8.	CONCLUSIONES Y LÍNEAS FUTURAS	124
8.1.	Conclusiones.....	124
8.2.	Líneas futuras	126
9.	BIBLIOGRAFÍA.....	128
	ANEXO I: MANUAL DE USUARIO	130
I. 1.	Registro de usuario e inicio de sesión.....	130
I. 2.	Eliminación de cuenta de usuario y edición de ajustes	132
I. 3.	Pantalla principal de la aplicación.....	134
I. 4.	Apertura y edición de ficheros	137
I. 5.	Creación de ficheros.....	139
I. 6.	Importación de ficheros	140
I. 7.	Borrado de ficheros.....	141
I. 8.	Visualización y adición de etiquetas	142
I. 9.	Eliminación de etiquetas.....	143
I. 10.	Filtrado de ficheros por etiquetas.....	144
I. 11.	Eliminación de etiqueta del filtro y desactivación del filtrado por etiqueta.....	145
I. 12.	Cierre de la aplicación y cierre de sesión de usuario	146

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Dropbox Vista General	25
Ilustración 2. Dropbox Vista Fotos	25
Ilustración 3. Google Drive Vista General	28
Ilustración 4. Google Drive Vista Reciente	28
Ilustración 5. Ubuntu One Files Vista General	30
Ilustración 6. Ubuntu One Files Menú Añadir.....	30
Ilustración 7. SkyDrive Vista General Lista	32
Ilustración 8. SkyDrive Vista General Mosaico.....	32
Ilustración 9. Diagrama de casos de uso	39
Ilustración 10. Arquitectura general del sistema	60
Ilustración 11. Diagrama de paquetes general	63
Ilustración 12. Diagrama de clases de la librería compartida	64
Ilustración 13. Diagrama de clases del cliente	65
Ilustración 14. Diagrama de clases del servidor.....	66
Ilustración 15. Diagrama de secuencia. Registrar usuario	68
Ilustración 16. Diagrama de secuencia. Iniciar sesión de usuario.....	69
Ilustración 17. Diagrama de secuencia. Importar fichero	70
Ilustración 18. Diagrama de secuencia. Abrir fichero	71
Ilustración 19. Diagrama de secuencia. Editar fichero.....	72
Ilustración 20. Diagrama de secuencia. Borrar fichero	73
Ilustración 21. Diagrama de secuencia. Modificar etiquetas.....	74
Ilustración 22. Diagrama de secuencia. Filtrar ficheros por etiqueta	75
Ilustración 23. Protocolo. Petición General	78
Ilustración 24. Protocolo. Respuesta General.....	78
Ilustración 25. Protocolo. Petición Signup	80
Ilustración 26. Protocolo. Respuesta Signup.....	80
Ilustración 27. Protocolo. Petición Accountdelete.....	80
Ilustración 28. Protocolo. Respuesta Accountdelete	81
Ilustración 29. Protocolo. Petición Login	81
Ilustración 30. Protocolo. Respuesta Login	82
Ilustración 31. Protocolo. Petición List.....	82
Ilustración 32. Protocolo. Respuesta List	83
Ilustración 33. Protocolo. Mensaje Date.....	85
Ilustración 34. Protocolo. Petición Get	86
Ilustración 35. Protocolo. Respuesta Get.....	86
Ilustración 36. Protocolo. Petición Put.....	87
Ilustración 37. Protocolo. Respuesta Put	87
Ilustración 38. Protocolo. Petición Update	88
Ilustración 39. Protocolo. Respuesta Update.....	88
Ilustración 40. Protocolo. Petición Labelupdate	90
Ilustración 41. Protocolo. Respuesta Labelupdate	90
Ilustración 42. Protocolo. Petición Delete.....	91

Ilustración 43. Protocolo. Respuesta Delete	92
Ilustración 44. Fichero de metadatos del servidor.....	94
Ilustración 45. Fichero de metadatos del cliente	96
Ilustración 46. Diagrama de Gantt	118
Ilustración 47. Menú de aplicaciones.....	130
Ilustración 48. Pantalla de bienvenida	130
Ilustración 49. Pantalla de registro	131
Ilustración 50. Pantalla de inicio de sesión	131
Ilustración 51. Menú de opciones de pantalla de bienvenida	132
Ilustración 52. Pantalla de eliminación de cuenta de usuario	132
Ilustración 53. Pantalla de ajustes.....	133
Ilustración 54. Diálogo de ajuste del puerto del servidor	133
Ilustración 55. Pantalla principal.....	135
Ilustración 56. Menú de opciones de la pantalla principal	135
Ilustración 57. Bandeja de notificaciones	136
Ilustración 58. Diálogo de descarga de fichero	137
Ilustración 59. Pantalla de elección de aplicación de apertura de fichero	137
Ilustración 60. Diálogo de apertura de fichero obsoleto	138
Ilustración 61. Pantalla de conflicto de versiones de fichero	138
Ilustración 62. Diálogo de creación de fichero.....	139
Ilustración 63. Diálogo de elección de aplicación para seleccionar fichero a importar.....	140
Ilustración 64. Diálogo de importación de fichero.....	140
Ilustración 65. Menú contextual de fichero	141
Ilustración 66. Pantalla de etiquetas.....	142
Ilustración 67. Diálogo de adición de etiqueta	142
Ilustración 68. Menú contextual de etiqueta.....	143
Ilustración 69. Filtrado por etiqueta "documentos"	144
Ilustración 70. Filtrado por etiquetas "documentos" y "privado"	144
Ilustración 71. Filtrado por etiqueta "privado"	145
Ilustración 72. Filtrado por etiquetas "fotos" y "privado"	145
Ilustración 73. Diálogo de cierre de sesión	146

ÍNDICE DE TABLAS

Tabla 1. Plantilla para Casos de Uso.....	40
Tabla 2. CU-01. Registrar usuario.....	41
Tabla 3. CU-02. Eliminar cuenta de usuario	42
Tabla 4. CU-03. Iniciar sesión de usuario	42
Tabla 5. CU-04. Cerrar sesión de usuario	43
Tabla 6. CU-05. Crear nuevo fichero	43
Tabla 7. CU-06. Importar fichero.....	44
Tabla 8. CU-07. Abrir fichero.....	45
Tabla 9. CU-08. Editar fichero	46
Tabla 10. CU-09. Borrar fichero.....	46
Tabla 11. CU-10. Mostrar etiquetas	46
Tabla 12. CU-11. Añadir etiqueta	47
Tabla 13. CU-12. Quitar etiqueta	47
Tabla 14. CU-13. Filtrar ficheros por etiqueta.....	48
Tabla 15. CU-14. Desactivar el filtro por etiqueta.....	48
Tabla 16. CU-15. Editar ajustes	49
Tabla 17. Plantilla para requisitos	50
Tabla 18. RF-01. Registro de usuario.....	52
Tabla 19. RF-02. Eliminación de cuenta de usuario	52
Tabla 20. RF-03. Inicio de sesión de usuario	52
Tabla 21. RF-04. Cierre de sesión de usuario	52
Tabla 22. RF-05. Creación de nuevo fichero	53
Tabla 23. RF-06. Importación de fichero.....	53
Tabla 24. RF-07. Apertura de fichero	53
Tabla 25. RF-08. Edición de fichero	53
Tabla 26. RF-09. Borrado de fichero.....	53
Tabla 27. RF-10. Visualización de la lista de ficheros	54
Tabla 28. RF-11. Visualización de etiquetas	54
Tabla 29. RF-12. Adición de etiqueta	54
Tabla 30. RF-13. Eliminación de etiqueta.....	54
Tabla 31. RF-14. Filtrado de ficheros por etiqueta.....	54
Tabla 32. RF-15. Filtrado de ficheros por etiqueta.....	55
Tabla 33. RF-16. Desactivación del filtrado por etiqueta	55
Tabla 34. RF-17. Edición de ajustes	55
Tabla 35. RNF-01. Descarga de ficheros a petición del usuario	56
Tabla 36. RNF-02. Sincronización asíncrona	56
Tabla 37. RNF-03. Comprobación de modificación de ficheros	56
Tabla 38. RNF-04. Control de versiones y conflictos de fichero.....	57
Tabla 39. RNF-05. Sincronización de lista de ficheros parcial	57
Tabla 40. RR-01. Conexión a internet.....	58
Tabla 41. RR-02. Gestor de ficheros	58
Tabla 42. RR-03. Aplicaciones externas para apertura de ficheros.....	58

Tabla 43. RR-04. Versión de Android	58
Tabla 44. Dispositivo de prueba 1	107
Tabla 45. Dispositivo de prueba 2	108
Tabla 46. Dispositivo de prueba 3	108
Tabla 47. Dispositivo de prueba 4	108
Tabla 48. Dispositivo de prueba 5	109
Tabla 49. Plantilla para Pruebas de Aceptación	110
Tabla 50. PA-01. Registro de usuario	110
Tabla 51. PA-02. Eliminación de cuenta de usuario	110
Tabla 52. PA-03. Inicio de sesión de usuario	111
Tabla 53. PA-04. Cierre de sesión de usuario	111
Tabla 54. PA-05. Creación de nuevo fichero	111
Tabla 55. PA-06. Importación de fichero	112
Tabla 56. PA-07. Apertura de fichero	112
Tabla 57. PA-08. Edición de fichero	112
Tabla 58. PA-09. Borrado de fichero	113
Tabla 59. PA-10. Visualización de la lista de ficheros	113
Tabla 60. PA-11. Visualización de etiquetas	113
Tabla 61. PA-12. Adición de etiqueta	113
Tabla 62. PA-13. Eliminación de etiqueta	114
Tabla 63. PA-14. Filtrado de ficheros por etiqueta	114
Tabla 64. PA-15. Filtrado de ficheros por etiqueta	114
Tabla 65. PA-16. Desactivación del filtrado por etiqueta	114
Tabla 66. PA-17. Edición de ajustes	115
Tabla 67. Matriz de trazabilidad Requisitos Funcionales - Pruebas	115
Tabla 68. Planificación de tareas del proyecto	117
Tabla 69. Costes de personal	120
Tabla 70. Costes de hardware	121
Tabla 71. Costes de software	122
Tabla 72. Coste total del proyecto	123

1. INTRODUCCIÓN

En este Proyecto Fin de Carrera se desarrollará una solución de almacenamiento distribuido que permita la distribución y sincronización de ficheros en varios dispositivos móviles que funcionen con la plataforma Android [1].

1.1.Motivación

El uso de herramientas para el almacenamiento, distribución y sincronización de ficheros se hace muy necesario hoy en día, debido a la gran cantidad de usuarios que disponen de varios dispositivos diferentes desde los cuales acceder a estos ficheros. Esto se hace más necesario si cabe teniendo en cuenta el gran auge de los dispositivos móviles tales como *smartphones* y *tablets*, de forma que el usuario pueda disponer de los ficheros que necesite en cualquier lugar que se encuentre.

Otra necesidad que tienen los usuarios es la de compartir ficheros entre diferentes usuarios, la cual puede ser cubierta con este tipo de herramientas. Esta necesidad se hace más evidente hoy en día debido a los entornos cada vez más colaborativos que encontramos tanto a nivel laboral como personal, impulsados por gran uso con que cuentan las redes sociales y la web en general y la gran cantidad de trabajadores que deben realizar su trabajo en equipo, tanto a nivel académico como empresarial.

Todo esto no sería posible sin el gran avance que han conseguido las tecnologías de computación móvil, con el desarrollo de sistemas operativos móviles de código abierto que permiten extender la funcionalidad a través de aplicaciones de terceros, así como de sus características hardware, que han aumentado la capacidad de almacenamiento de los dispositivos móviles, y su potencia de procesamiento (principalmente smartphones y tablets). Además, también ha sido necesario el desarrollo de las redes de comunicaciones móviles que han aumentado la velocidad y la disponibilidad de la conexión a internet (principalmente 3G y 4G).

1.2.Objetivos

El principal objetivo de este Proyecto Fin de Carrera es desarrollar un sistema de almacenamiento, distribución y sincronización de ficheros que los usuarios puedan utilizar en dispositivos que funcionen con el sistema operativo Android.

Para la consecución de este objetivo principal, existen los siguientes objetivos secundarios a cumplir:

- Estudio y elección de las tecnologías más adecuadas con las que implementar el sistema.
- Análisis de los sistemas similares que se encuentran en el mercado.
- Desarrollo de un sistema de almacenamiento de los ficheros de los usuarios.
- Desarrollo de un sistema de procesado y almacenamiento de los metadatos de los ficheros de los usuarios.
- Desarrollo de un sistema de etiquetas que permita a los usuarios etiquetar e identificar los ficheros y navegar por los ficheros usando estas etiquetas.
- Desarrollo de un protocolo de comunicación entre el cliente y el servidor del sistema.
- Desarrollo de una aplicación cliente para el sistema que incluya una interfaz gráfica de usuario que sea sencilla y fácil de usar.
- Desarrollo de una aplicación servidor para el sistema que responda a las peticiones de las aplicaciones clientes.

1.3.Estructura de la memoria

Este documento se encuentra estructurado en capítulos ordenados de forma lógica, de tal manera que su lectura en orden permita una comprensión más sencilla del mismo.

A continuación se presenta una breve descripción de cada uno de los capítulos de este documento.

- 1. Introducción:** Es el capítulo introductorio de la memoria que pretende dar una visión general del proyecto. Incluye la motivación y los objetivos del proyecto, así como un glosario de términos utilizados en este documento.
- 2. Estado del arte:** En este capítulo se describirán las tecnologías en las que está basado el sistema desarrollado y se realizará un análisis de

varias soluciones o herramientas existentes en el mercado similares al sistema desarrollado en este proyecto.

3. **Análisis del sistema:** En este capítulo se va a definir cuál es el problema que queremos solucionar con este sistema mediante una descripción general del mismo. Después se va a describir cuál va a ser la funcionalidad del sistema mediante una serie de casos de uso y de requisitos.
4. **Diseño del sistema:** En este capítulo se va a describir el diseño del sistema. En primer lugar se definirá el diseño de una forma más general mediante la descripción de su arquitectura y después se especificará de una forma más detallada mediante diagramas de clases y de secuencia.
5. **Implementación del sistema:** En este capítulo se explican los aspectos relativos a la implementación del sistema. Para ello se describe el protocolo de comunicación entre cliente y servidor, el formato de los ficheros utilizados en el sistema, la codificación de los datos utilizada, los sockets e hilos empleados y otras consideraciones de la implementación.
6. **Pruebas:** En este capítulo se van a especificar las pruebas de aceptación realizadas al sistema desarrollado para comprobar que funciona correctamente y se cumplen todos los requisitos especificados.
7. **Planificación y presupuesto:** En este capítulo se va a especificar en primer lugar la planificación del proyecto y después el presupuesto requerido para la realización del mismo.
8. **Conclusiones y líneas futuras:** En este capítulo se presentan las conclusiones extraídas del desarrollo de este proyecto, así como las posibles líneas futuras de desarrollo del mismo.
9. **Bibliografía:** En este capítulo se presentan todas las fuentes de información consultadas para la realización del proyecto.

1.4.Glosario de términos

- **ADT:** Android Development Toolkit (herramientas de desarrollo de Android). Es el conjunto de herramientas proporcionadas por Google para facilitar el desarrollo de aplicaciones Android.
- **Android:** Sistema operativo para dispositivos móviles desarrollado por Google.

- **BlackBerry:** Sistema operativo para dispositivos móviles desarrollado por RIM.
- **Gmail:** Servicio de correo electrónico de Google.
- **Hash:** Es una función matemática que permite proporcionar un resumen de un conjunto de datos. Se utiliza ampliamente en criptografía, generalmente para asegurar la integridad de los datos.
- **HTTP:** Hypertext Transfer Protocol (protocolo de transferencia de hipertexto). Protocolo diseñado para la transferencia de hipertexto a través de una conexión. Es una parte fundamental de la World Wide Web.
- **iOS:** Sistema operativo para dispositivos móviles desarrollado por Apple.
- **IP:** Internet Protocol (Protocolo de Internet). Es el protocolo que posibilita la interconexión de distintas redes de ordenadores o dispositivos informáticos.
- **IVA:** Impuesto sobre el valor añadido.
- **Kindle Fire:** Tableta desarrollada por Amazon con sistema operativo Android.
- **Linux:** Sistema operativo para ordenadores de código abierto.
- **Mac:** Sistema operativo para ordenadores desarrollado por Apple.
- **PC:** Personal Computer (Ordenador personal).
- **Plug-in:** Extensión o complemento de una aplicación que le aporta una nueva funcionalidad.
- **REST:** Representational State Transfer. Es un conjunto de técnicas de arquitectura software para sistemas hipermedia distribuidos.
- **SD:** Secure Digital. Es uno de los formatos de tarjeta de memoria más extendidos.
- **Smartphone:** Dispositivo móvil similar a un teléfono móvil que además de funciones de telefonía ofrece prestaciones similares a un ordenador. Típicamente cuentan con una pantalla táctil.
- **Socket:** Es una conexión entre dos programas que les permite intercambiar datos. Típicamente esta conexión se realiza a través de internet para comunicar dos ordenadores o dispositivos diferentes.
- **SSL:** Secure Socket Layer (capa de conexión segura). Es un conjunto de protocolos criptográficos que proporcionan comunicaciones seguras a través de la red.

- **Tablet:** Dispositivo móvil que ofrece prestaciones similares a un ordenador pero más ligero y pequeño. Típicamente no tienen teclado ni ratón y en su lugar cuentan con una pantalla táctil. Son similares a un smartphone pero de mayor tamaño y no suelen incorporar funciones de telefonía.
- **TCP:** Transmission Control Protocol (Protocolo de Control de Transmisión), es un protocolo que pertenece a la capa de transporte del conjunto de protocolos TCP/IP, y es uno de los protocolos fundamentales en Internet. Permite asegurar una conexión fiable y ordenada.
- **TLS:** Transport Layer Security (seguridad de la capa de transporte). Es el sucesor de SSL y proporciona algunos avances respecto a éste.
- **WiFi:** Mecanismo de conexión de dispositivos electrónicos de forma inalámbrica, usado generalmente para conectar dispositivos a internet a través de un punto de acceso inalámbrico. Se utilizan principalmente para redes de área local, es decir, dentro de un edificio (un hogar o un edificio de oficinas o comercial) o en un área reducida como el campus de una universidad.
- **Windows:** Sistema operativo para ordenadores personales desarrollado por Microsoft.
- **World Wide Web:** Sistema de hipertexto o hipermedios interconectados y accesibles a través de Internet.
- **XML:** Extensible Markup Language. Es un lenguaje de marcado extensible que permite especificar otros lenguajes.

2. ESTADO DEL ARTE

En este capítulo se describirán las tecnologías en las que está basado el sistema desarrollado. Por una parte describiremos la plataforma Android, que es el sistema operativo que poseen los dispositivos móviles donde se ejecutará el cliente del sistema. Después describiremos Java [2], que es el lenguaje de programación en el que se implementará tanto el cliente como el servidor. Además se describirá Eclipse [3], que es el entorno de desarrollo utilizado para implementar y probar el sistema desarrollado. También se describirá XML [4], que es un lenguaje de marcado que se utiliza cuando se desarrolla una aplicación en Android para especificar algunas cosas como configuraciones, interfaces gráficas de usuario y otros valores necesarios para la aplicación.

Finalmente se realizará también un análisis de varias soluciones o herramientas existentes en el mercado similares al sistema desarrollado en este proyecto, atendiendo principalmente al funcionamiento de sus aplicaciones clientes para Android.

2.1.Android

Android es un sistema operativo para dispositivos móviles con pantalla táctil (principalmente smartphones y tablets), basado en Linux. Inicialmente fue desarrollado por Android, Inc. y Google lo compró en 2005. Android fue presentado en 2007 junto con la fundación del Open Handset Alliance [5], un consorcio de compañías de hardware, software y telecomunicaciones que impulsan el desarrollo de estándares abiertos para los dispositivos móviles. El primer dispositivo con sistema operativo Android fue vendido en octubre de 2008.

Android es un sistema operativo de código abierto liberado por Google bajo la licencia Apache License [6], lo que posibilita que tanto los fabricantes de dispositivos y operadores de telefonía como muchos entusiastas puedan modificar el código libremente. Además, Android cuenta con una gran comunidad de desarrolladores de aplicaciones que extienden la funcionalidad de los dispositivos. Este tipo de aplicaciones, están implementadas en una versión modificada de Java y se distribuyen a los usuarios a través de Google Play [7], que es la tienda en línea de Google.

Todos estos factores hacen que hoy en día Android sea la plataforma para smartphones más usada a nivel mundial. Aunque principalmente esté diseñada para smartphones y tablets, actualmente es usado por una gran cantidad de dispositivos de otros tipos como televisores, consolas de videojuegos o cámaras de fotos digitales.

Precisamente, una de las características de Android es la gran variedad de características de los dispositivos, que pueden tener diferente hardware (diferentes tamaños y resoluciones de pantalla, diferentes sistemas de hardware como GPS, acelerómetro, cámara, etc.) y diferentes versiones de Android instaladas. Esto hace algo más complicado el diseño e implementación de las aplicaciones para que se adapten adecuadamente a las características de cada dispositivo, aunque el propio Android facilita esta labor.

Otra característica muy interesante de Android, principalmente para los desarrolladores de aplicaciones, es que permite que las aplicaciones puedan usar funcionalidades proporcionadas por otras aplicaciones que estén instaladas en el dispositivo, no solamente las funcionalidades y librerías aportadas por el propio sistema operativo. De esta forma las aplicaciones desarrolladas pueden disponer de funcionalidades desarrolladas por terceros y se evitan tener que implementar dicha funcionalidad por sí mismas, facilitando la reutilización de código.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un *framework* Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución.

Android pone a disposición de los desarrolladores una colección de herramientas de desarrollo (Android Developer Tools [8]) que además se integran con el entorno de desarrollo Eclipse que se explicará en la sección correspondiente, ofreciendo muchas herramientas y facilidades para el desarrollo de una forma centralizada.

2.2.Java

Java es un lenguaje de programación de propósito general desarrollado por Sun Microsystems y posteriormente adquirido por Oracle. Es un lenguaje multiplataforma ya que se compila en un *bytecode* que se puede ejecutar en cualquier Máquina Virtual de Java, lo que independiza los programas desarrollados con él de la arquitectura de plataforma subyacente. Esto convierte el código en portable, de forma que se pueda implementar una vez y poder ejecutar en cualquier plataforma sin tener que modificar el código.

Además es un lenguaje de alto nivel, concurrente y orientado a objetos. Esto último quiere decir que el código se divide en clases que cuentan con métodos y atributos. Cuando se ejecuta un programa en Java, se crean instancias de las clases, es decir, objetos. Es un lenguaje que permite la herencia, con lo que se pueden establecer

jerarquías entre las clases y además permite polimorfismo, de forma que varias clases derivadas dentro de la jerarquía puedan implementar de forma distinta un mismo método.

Es un lenguaje que cuenta con una gestión de memoria muy depurada. No permite el uso de punteros, en su lugar usa referencias a los diferentes objetos, lo que evita errores con su manejo. Además cuenta con un recolector de basura que destruye los objetos automáticamente cuando ya no van a ser más usados para liberar memoria y así evitar fugas como puede ocurrir en otros lenguajes donde la liberación de memoria se realiza de forma manual.

Además de la máquina virtual de Java, o entorno de ejecución de Java (Java Runtime Environment) que permite ejecutar los programas escritos en Java, los desarrolladores además cuentan con un conjunto de herramientas de desarrollo (Java Development Kit) para facilitar las tareas de desarrollo, depuración y monitorización de aplicaciones Java.

Java también cuenta con una gran cantidad de librerías disponibles, lo que permite aumentar la funcionalidad de las aplicaciones desarrolladas además de evitar una gran cantidad de trabajo a los desarrolladores. Además cuenta con numerosas herramientas y entornos de desarrollo ofrecidos por terceros, algunos de ellos muy completos que facilitan enormemente esta labor.

2.3.Eclipse

Eclipse es un entorno de desarrollo integrado (IDE son sus siglas en inglés) multiplataforma y de código abierto. Facilita el desarrollo de software integrando todas las herramientas de desarrollo en una sola aplicación, que además cuenta con una interfaz gráfica integrada que permite utilizar todas las herramientas de forma sencilla. Eclipse consta de un espacio de trabajo base que es extensible, de forma que mediante diferentes *plug-ins* se consigue adaptar el entorno para el desarrollo de software en diferentes lenguajes o tecnologías, además de incluir numerosas herramientas adicionales. En este sentido se podría considerar que Eclipse es más bien una plataforma de desarrollo de entornos de desarrollo integrados ya que no se limita a ser un único entorno.

El IDE de desarrollo Java es seguramente la versión o paquete más conocido y utilizado de Eclipse. En este proyecto se ha utilizado el paquete ADT (Android Developer Tools), proporcionado por Google y que está diseñado específicamente para el desarrollo de aplicaciones Android. Dicho paquete incluye Eclipse con el plug-in ADT además de otras herramientas adicionales para desarrollo en Android.

Eclipse consta de un núcleo de ejecución y sobre este emplea plug-ins para proporcionar todas sus funcionalidades, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de plug-ins es una plataforma ligera para componentes de software.

Eclipse cuenta también con una comunidad muy activa de usuarios que desarrollan nuevos plug-ins que extienden la funcionalidad de Eclipse cubriendo nuevas áreas de aplicación.

2.4.XML

XML (Extensible Markup Language) es un lenguaje de marcado extensible creado por el World Wide Web Consortium (W3C) [9]. En realidad podríamos definirlo como un metalenguaje ya que precisamente su característica de ser extensible permite utilizarlo para definir otros lenguajes. Es un lenguaje en el que la información está codificada de forma legible.

XML es un lenguaje derivado de SGML, que es un lenguaje más extenso y complejo, es decir, XML es un subconjunto de SGML. XML se diseñó para facilitar la implementación del analizador sintáctico respecto de SGML, que resulta mucho más complejo. Para ello se simplificó eliminando algunas opciones y características de SGML además de imponer algunas restricciones a su sintaxis original. Una de las ventajas de XML es que el mismo analizador sintáctico se puede usar para cualquier lenguaje definido con XML, es decir, no es necesario definir un nuevo analizador específico para cada nueva versión definida.

XML fue diseñado originalmente para cubrir las necesidades de la publicación electrónica a gran escala, aunque actualmente tiene un papel muy importante en el intercambio de información estructurada, incluso entre plataformas y sistemas diferentes. Se ha utilizado para definir gran cantidad de formatos de almacenamiento de información utilizados en aplicaciones como procesadores de texto, hojas de cálculo, editores de imagen, etc. También se ha utilizado como base de una gran cantidad de protocolos de comunicación.

La unidad de información de XML se denomina “documento XML”. En XML se presenta la información de forma estructurada en unidades bien definidas llamadas elementos. En los documentos XML estos elementos se indican con etiquetas que incluyen un nombre, que es el que indica el tipo de elemento, además de poder tener atributos. De esta manera se presenta la estructura lógica del documento, que toma forma de árbol. Además, XML permite la declaración del tipo de documento, de manera que se especifica la gramática que debe seguir ese tipo de documentos, es

decir, la estructura que deben presentar, incluyendo los elementos que pueden contener y la forma en que estos se anidan, así como sus atributos.

2.5. Análisis de soluciones en el mercado

El objetivo de esta sección es encontrar y analizar distintas soluciones o herramientas de almacenamiento, distribución y sincronización de ficheros existentes en el mercado, atendiendo principalmente a la funcionalidad que presentan los clientes en la plataforma de computación móvil Android, aunque también puedan funcionar en otras plataformas diferentes, tanto móviles como no, y en la web. Para ello se tendrán en cuenta diferentes aspectos como son la usabilidad de cada herramienta, las funcionalidades y las prestaciones que ofrece cada una.

Para elegir estas soluciones se ha tenido en cuenta sobre todo su popularidad, es decir, se ha intentado elegir soluciones que tengan bastante aceptación por parte de los usuarios así como las que han sido desarrolladas por compañías de gran importancia en el sector de la informática. Además, como requisito imprescindible las aplicaciones seleccionadas deben funcionar en la plataforma Android, como ya se ha dicho anteriormente. Con ello se intenta que las aplicaciones seleccionadas sean bastante conocidas por el público.

Las aplicaciones seleccionadas son las siguientes:

- Dropbox [10]
- Google Drive [11]
- Ubuntu One Files [12]
- Microsoft SkyDrive [13]

Para cada una de ellas se tendrán en cuenta, entre otros, los siguientes aspectos principales:

- Usabilidad
- Prestaciones:
 - Capacidad de almacenamiento
 - Plataformas soportadas
- Funcionalidades:
 - Acceso a través de la Web
 - Soporte de directorios

- Carga automática de fotografías desde el dispositivo
- Gestión de fotos (álbumes)
- Compartir ficheros y directorios con otros usuarios
- Ficheros sin conexión en dispositivo móvil
- Búsqueda y filtrado de ficheros y directorios
- Ordenación de ficheros y directorios visualizados
- Historial de eventos
- Historial de versiones de ficheros
- Apertura nativa de ficheros
- Edición de ficheros
- Importar ficheros existentes en el dispositivo
- Crear, mover, borrar y renombrar ficheros y directorios

En las siguientes subsecciones se detallan cada una de las aplicaciones, incluyendo capturas de pantalla para cada una de ellas que ayuden a ejemplificar el aspecto de la interfaz de usuario.

2.5.1. Dropbox

Es probablemente la aplicación de este tipo más conocida y la primera que consiguió cierta notoriedad. Actualmente ofrece sus servicios a través de aplicaciones para ordenadores de las principales plataformas (Windows, Linux y Mac), para dispositivos móviles (Android, iOS, BlackBerry y Kindle Fire) y a través de la web. La cuenta básica gratuita ofrece una capacidad de almacenamiento inicial de 2GB que pueden ir aumentando hasta los 18GB gracias a las recomendaciones a otras personas.

Ofrece un historial de eventos que muestra cronológicamente todo lo que ha ocurrido en la cuenta y que incluye además un historial de versiones de ficheros que permite ver todas las modificaciones realizadas a cada fichero y recuperar cualquier versión anterior. Esta funcionalidad sólo está disponible a través de la web en versión de escritorio.

Cliente Dropbox para Android

El cliente para Android de Dropbox permite acceder a todos los ficheros y directorios de la cuenta del usuario a través de su dispositivo Android. Consiste en una aplicación muy sencilla de utilizar para el usuario. Por defecto los ficheros se descargan al dispositivo bajo petición del usuario, por lo que no es posible acceder a ellos sin conexión a internet, aunque permite al usuario marcar determinados archivos como favoritos para que se sincronicen en el dispositivo y que estén disponibles sin conexión.

Además permite subir ficheros existentes en el dispositivo, así como crear ficheros y directorios nuevos, moverlos, eliminarlos y cambiarles el nombre. Permite además la opción de subir automáticamente las fotografías sacadas con la cámara del dispositivo y configurar si ésta se realizará únicamente cuando el dispositivo se conecta a internet por WiFi o también a través de la red de datos móvil (con lo que se consigue ahorrar tráfico en ésta última si se selecciona solo WiFi). Además ofrece una interfaz específica para gestionar las fotografías que las muestra ordenadas por fecha y permite clasificarlas en álbumes.

Además permite ordenar los ficheros y directorios visibles alfabéticamente o por fecha además de hacer búsquedas de ambos por nombre. También permite compartir tanto ficheros como directorios con otros usuarios y exportar los ficheros con otras aplicaciones.

Esta aplicación permite abrir ficheros de los tipos básicos de forma nativa y de otros tipos de ficheros mediante otras aplicaciones externas. En cuanto a la edición,

aunque sólo permite editar ficheros de texto plano de forma nativa, también permite editarlos con aplicaciones externas, y los cambios realizados se sincronizan correctamente en la cuenta. Como se ha comentado anteriormente sólo se puede acceder al historial de eventos y versiones a través de la web, por lo que el cliente para Android no ofrece esta funcionalidad.

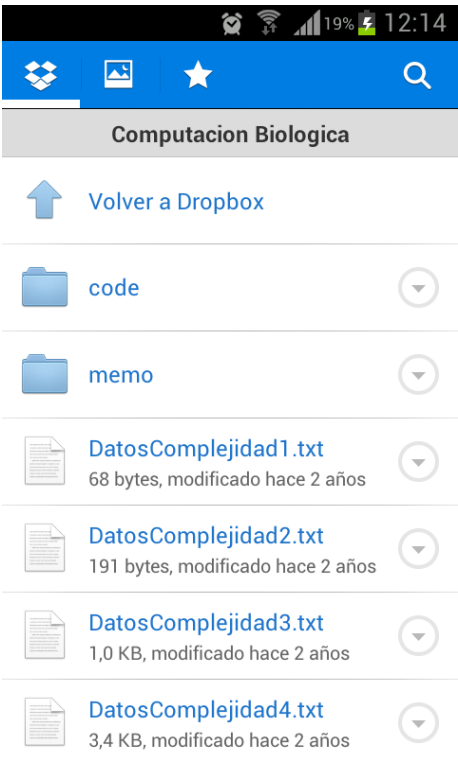


Ilustración 1. Dropbox Vista General

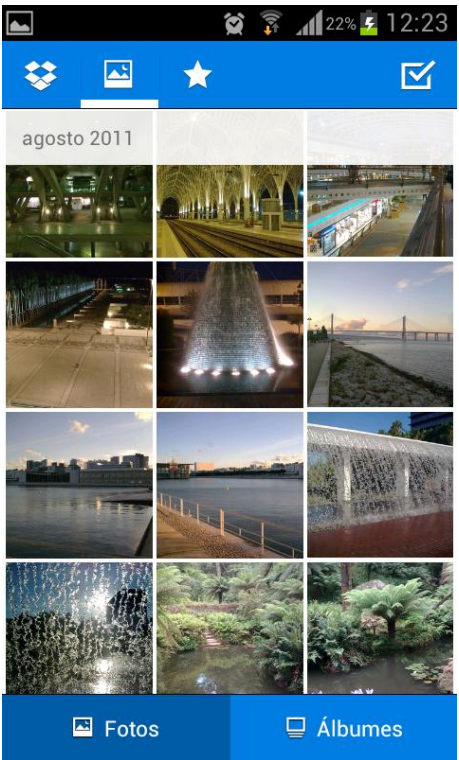


Ilustración 2. Dropbox Vista Fotos

2.5.2. Google Drive

Google Drive es una aplicación desarrollada por Google, la que es seguramente la compañía más conocida de internet, principalmente gracias a su buscador web, la cual es además la desarrolladora de Android. Esta aplicación ha pasado a sustituir a Google Docs de forma que integra toda la funcionalidad de este último, como la capacidad de visualizar diferentes tipos de documentos en línea, como documentos de texto formateado y hojas de cálculo, así como de permitir su edición de forma colaborativa entre varios usuarios. Google Drive añade a todo esto las capacidades de poder sincronizar los ficheros y directorios entre varios dispositivos.

Actualmente ofrece sus servicios a través de aplicaciones clientes para Windows, Mac, Android, iOS y a través de la Web. Las cuentas de Google Drive ofrecen una capacidad de almacenamiento de 15 GB en su versión gratuita. Además, los documentos guardados en formato Google Docs no ocupan espacio en la cuenta. Cuenta con la gran ventaja de que se integra con Gmail, al menos la versión web, de forma que permite añadir documentos adjuntos directamente a la cuenta de Google Drive del usuario.

Ofrece un historial de versiones de ficheros el cual permite ver todas las modificaciones realizadas a cada fichero y recuperar cualquier versión anterior hasta un máximo de 30 días o 100 versiones. Además ofrece un historial de eventos, pero a diferencia de Dropbox es mucho menos detallado ya que muestra únicamente las modificaciones en ficheros y no se especifica lo ocurrido con los directorios o la modificación que se ha llevado a cabo, es decir, si se ha creado un elemento nuevo, se ha editado, se ha borrado, etc. Estas funcionalidades de historial sólo están disponibles a través de la web en versión de escritorio, en la versión para Android sólo se muestra un historial de los últimos archivos abiertos.

Cliente Google Drive para Android

El cliente para Android de Google Drive, al igual que Dropbox, permite acceder a todos los ficheros y directorios de la cuenta del usuario y éstos se descargan bajo petición, aunque permite al usuario marcar determinados archivos para que se sincronicen en el dispositivo y que estén disponibles sin conexión. Además permite subir ficheros existentes en el dispositivo, así como crear ficheros y directorios nuevos, moverlos, eliminarlos y cambiarles el nombre. Es una aplicación sencilla de utilizar para el usuario aunque quizá algo menos intuitiva que Dropbox.

Esta aplicación no permite la opción de subir automáticamente las fotografías sacadas con la cámara del dispositivo ni ofrece una interfaz específica para gestionar las fotografías y clasificarlas en álbumes. Sí permite configurar si la sincronización de los archivos sin conexión se realizará únicamente cuando el dispositivo se conecta a internet por WiFi o también a través de la red de datos móvil (con lo que se consigue ahorrar tráfico en ésta última si se selecciona solo WiFi).

Además ofrece unas funcionalidades de ordenación y filtrado más avanzadas que Dropbox ya que permite ordenar los ficheros y directorios visibles alfabéticamente y por fecha de modificación, apertura y cambio realizado por el propio usuario. Además, permite filtrar por tipo de documento y realizar búsquedas por nombre. También permite compartir los ficheros y directorios con otras aplicaciones y guardarlos en el dispositivo.

Esta aplicación permite abrir ficheros de los tipos básicos y de Google Docs de forma nativa y de otros tipos de ficheros mediante otras aplicaciones externas. En cuanto a la edición de ficheros, sólo permite editar los ficheros de Google Docs de forma nativa, y la edición con aplicaciones externas no modifica el fichero original, ya que al guardar los cambios hay que elegir una nueva ubicación para el fichero, lo cual es un gran inconveniente que limita bastante la utilidad de la aplicación. Como se ha comentado anteriormente sólo se puede acceder al historial de eventos y versiones a través de la web, por lo que el cliente para Android sólo ofrece una lista de los últimos archivos abiertos, llamada reciente.

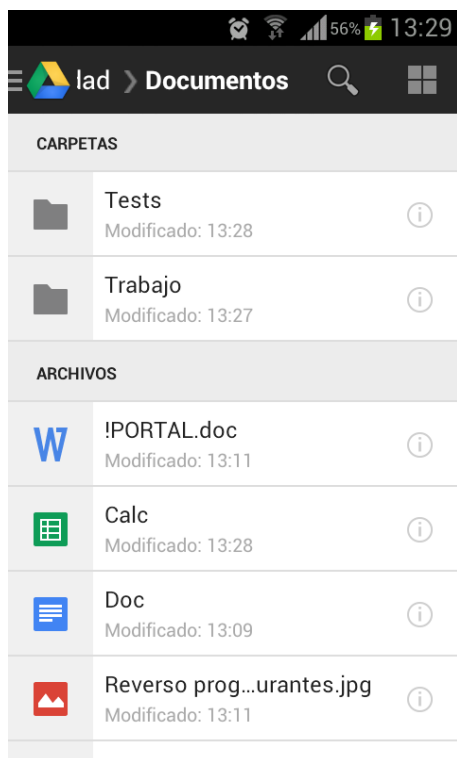


Ilustración 3. Google Drive Vista General

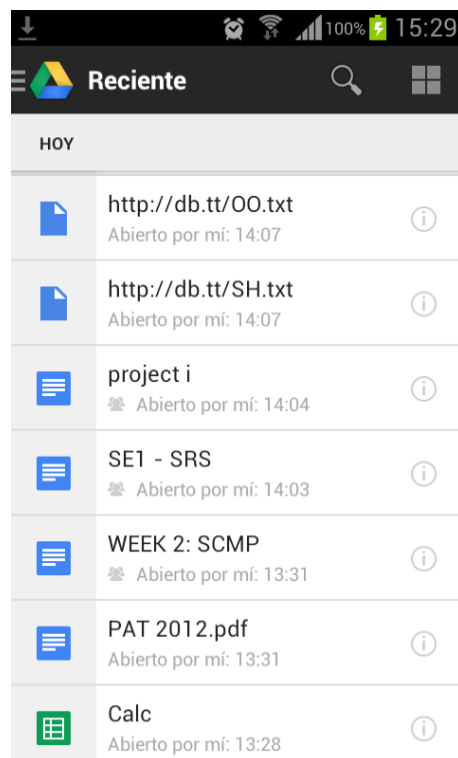


Ilustración 4. Google Drive Vista Reciente

2.5.3. Ubuntu One Files

Ubuntu One Files es una aplicación desarrollada por Canonical, una compañía cuyo producto más conocido es Ubuntu, un sistema operativo con núcleo Linux basado en Debian, que es seguramente el sistema operativo gratuito y de código abierto más popular hoy en día. Ubuntu One Files se integra en el sistema operativo Ubuntu, aunque a diferencia de éste, parte de su servidor es propietario. Actualmente ofrece sus servicios a través de clientes para Windows, Linux, Mac, Android, iOS y a través de la web. La cuenta gratuita ofrece 5GB de capacidad de almacenamiento que pueden aumentar si se invita a amigos.

A diferencia de las anteriores, esta aplicación no proporciona ni historial de eventos ni historial de versiones de ficheros.

Cliente Ubuntu One Files para Android

El cliente para Android de Ubuntu One Files permite acceder a los ficheros y directorios en la cuenta del usuario. La aplicación es mucho más simple que las anteriores ya que ofrece mucha menos funcionalidad. Al igual que las anteriores los ficheros se descargan bajo petición del usuario, por lo que no es posible acceder a ellos sin conexión a internet. A diferencia de las aplicaciones anteriores ésta no permite al usuario marcar determinados archivos como favoritos para que se sincronicen en el dispositivo y estén disponibles sin conexión.

Permite subir ficheros existentes en el dispositivo, así como crear directorios nuevos, pero no crear ficheros nuevos. Permite eliminar y cambiar el nombre tanto a ficheros como a directorios, pero no permite moverlos.

Al igual que Dropbox, también ofrece la posibilidad de subir automáticamente las fotografías sacadas con la cámara del dispositivo, pero no cuenta con una interfaz específica para gestionar las fotografías y clasificarlas en álbumes. Sí permite configurar si la sincronización de las fotos se realizará únicamente cuando el dispositivo se conecta a internet por WiFi o también a través de la red de datos móvil (con lo que se consigue ahorrar tráfico en ésta última si se selecciona solo WiFi).

A diferencia de las anteriores, esta aplicación no ofrece ninguna funcionalidad de ordenación, filtrado, ni búsqueda de los ficheros y directorios en la interfaz gráfica, pero sí permite guardar los ficheros en el dispositivo y compartir los ficheros con otros usuarios, aunque no permite compartir directorios.

Al igual que Dropbox esta aplicación permite abrir ficheros de los tipos más básicos de forma nativa y de otros tipos de ficheros mediante otras aplicaciones externas. En cuanto a la edición de ficheros, esta aplicación es mucho peor que las anteriores ya que sólo permite editar los ficheros con aplicaciones externas y los cambios realizados no se sincronizan en la cuenta, lo cual es un gran inconveniente que limita enormemente la utilidad de la aplicación. Como se ha dicho anteriormente no se ofrece ningún tipo de historial de eventos ni de versiones de ficheros.

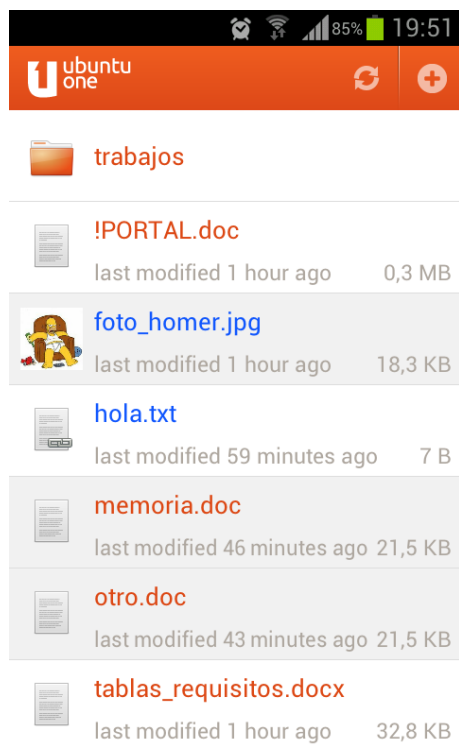
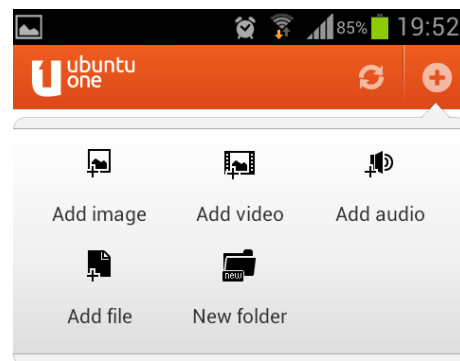


Ilustración 5. Ubuntu One Files Vista General



Loading files...

Ilustración 6. Ubuntu One Files Menú Añadir

2.5.4. Microsoft SkyDrive

SkyDrive es una aplicación desarrollada por Microsoft, una de las principales compañías de software del mundo, conocida principalmente por el sistema operativo Windows y la suite ofimática Microsoft Office.

Esta aplicación ofrece sus servicios a través de clientes para Windows, Mac, Android, iOS, Windows Phone y a través de la Web, aunque curiosamente, al contrario que las anteriores, no dispone de aplicación para Windows XP (únicamente funciona con Windows Vista o posteriores). La cuenta gratuita proporciona 7GB de espacio de almacenamiento.

Al contrario que Dropbox y Google Drive, esta aplicación no proporciona historial de versiones de los ficheros ni historial de eventos. Únicamente se muestra una lista de los archivos recientes, igual que pasa con el cliente de Android de Google Drive.

Cliente SkyDrive para Android

El cliente para Android de SkyDrive, al igual que todos los anteriores, permite acceder a todos los ficheros y directorios de la cuenta del usuario. Éstos se descargan bajo petición del usuario por lo que no es posible acceder a ellos sin conexión a internet. A diferencia de Dropbox y Google Drive, esta aplicación no permite al usuario marcar determinados archivos como favoritos para que se guarden localmente y estén disponibles sin conexión. La aplicación, de forma similar a Ubuntu One Files, ofrece mucha menos funcionalidad que Dropbox y Google Drive. Al igual que en todas ellas, la interfaz es bastante sencilla de utilizar para el usuario.

Igual que Ubuntu One Files, permite subir ficheros existentes en el dispositivo, así como crear directorios nuevos, pero no crear ficheros nuevos. Permite eliminar y cambiar el nombre tanto a ficheros como a directorios, pero no permite moverlos. Tampoco proporciona la opción de subir automáticamente las fotografías sacadas con la cámara del dispositivo ni ofrece una interfaz específica para gestionar las fotografías y clasificarlas en álbumes

De forma similar a Ubuntu One Files, no incorpora ninguna función para ordenar o filtrar los ficheros y directorios visibles ni para hacer búsquedas. En cambio sí permite compartir tanto ficheros como directorios con otros usuarios y guardar los ficheros en el dispositivo.

Esta aplicación permite abrir ficheros de los tipos más básicos de forma nativa y de otros tipos de ficheros mediante aplicaciones externas. En cuanto a la edición de ficheros, esta aplicación deja bastante que desear ya que sólo permite editar los ficheros con aplicaciones externas, y de igual forma que ocurre con Google Drive, no modifica el fichero original, ya que al guardar los cambios hay que elegir una nueva ubicación para el fichero, lo cual es un gran inconveniente que limita enormemente la utilidad de la aplicación. Como se ha dicho anteriormente no se ofrece ningún tipo de historial de eventos ni de versiones de ficheros, únicamente se ofrece una lista de documentos recientes.

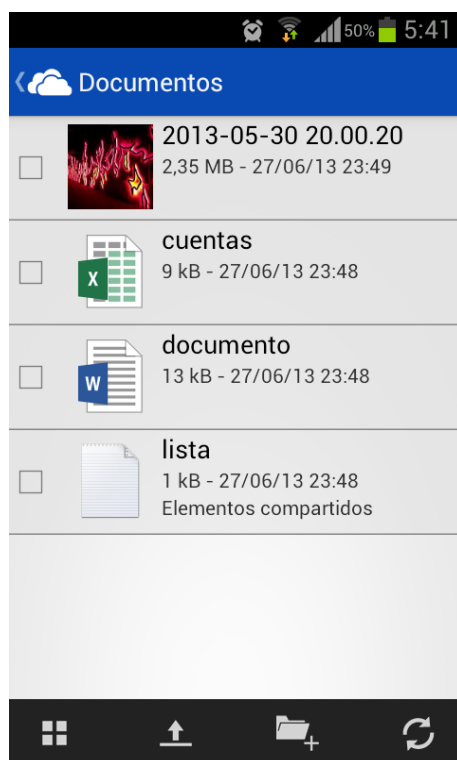


Ilustración 7. SkyDrive Vista General Lista

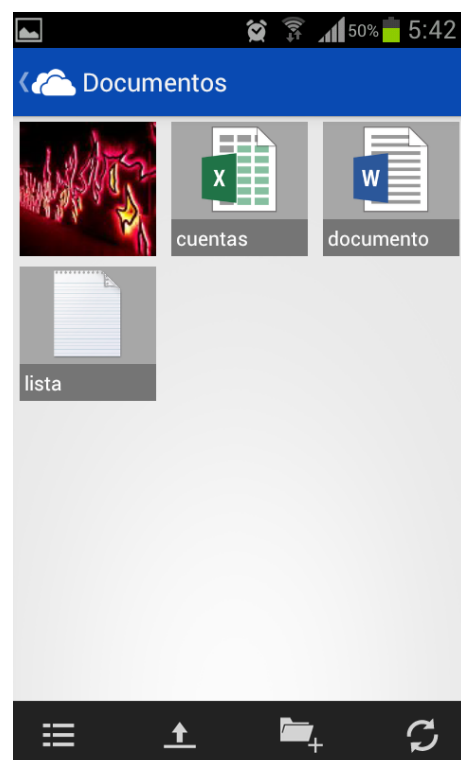


Ilustración 8. SkyDrive Vista General Mosaico

2.5.5. Resumen de características de las soluciones analizadas

A continuación se presenta una tabla resumen de las características de cada aplicación analizada, organizadas de tal manera que se facilite la comparación de todas ellas y permita extraer conclusiones más fácilmente, las cuales serán presentadas en el siguiente apartado.

Para facilitar la comprensión y la comparación se ha utilizado el siguiente código de colores:

Característica positiva
Característica intermedia
Característica negativa

Servicio	Dropbox	Google Drive	Ubuntu One Files	Microsoft SkyDrive
Usabilidad	Muy Buena	Buena	Buena	Buena
Capacidad	2GB y puede aumentar hasta 18 GB	15 GB	5GB y puede aumentar	7GB
Plataformas fijas	Windows, Linux y Mac	Windows y Mac	Windows, Linux y Mac	Windows (no XP) y Mac
Plataformas móviles	Android, iOS, BlackBerry y Kindle Fire	Android e iOS	Android e iOS	Android, iOS y Windows Phone
Acceso Web	Sí	Sí	Sí	Sí
Directorios	Sí	Sí	Sí	Sí
Carga automática de fotos	Sí	No	Sí	No
Gestión de fotos	Sí. Con álbumes	No	No	No
Compartir con otros usuarios	Sí	Sí	Sólo ficheros	Sí
Ficheros sin conexión	Sólo los marcados	Sólo los marcados	No	No
Búsqueda	Sí	Sí	No	No

Servicio	Dropbox	Google Drive	Ubuntu One Files	Microsoft SkyDrive
Ordenación en la vista	Alfabética y por fecha	Alfabética y por fecha de modificación, apertura y cambio propio. Permite filtrar por tipo de documento	No	No
Historial de eventos	Sólo en Web. Muy detallado	Sólo en Web. Poco detallado	No	No
Historial de ficheros	Sólo en Web	Sólo en Web	No	No
Apertura nativa de distintos tipos de ficheros	Bastante limitada	Incluye documentos de Google Docs	Bastante limitada	Bastante limitada
Edición de ficheros	Nativa bastante limitada. Sincroniza los cambios al editar con otras aplicaciones	Nativa incluye documentos de Google Docs. No permite editar a través de otras aplicaciones	No cuenta con edición nativa. No sincroniza los cambios al editar con otras aplicaciones	No cuenta con edición nativa. No permite editar a través de otras aplicaciones
Importar ficheros existentes en el dispositivo	Sí	Sí	Sí	Sí
Crear nuevos	Ficheros y directorios	Ficheros y directorios	Directorios	Directorios
Mover	Ficheros y directorios	Ficheros y directorios	No	No
Borrar	Ficheros y directorios	Ficheros y directorios	Ficheros y directorios	Ficheros y directorios
Renombrar	Ficheros y directorios	Ficheros y directorios	Ficheros y directorios	Ficheros y directorios

2.5.6. Conclusiones sobre las soluciones analizadas

A la vista de los resultados del análisis de las distintas aplicaciones, podemos extraer las siguientes conclusiones.

Aunque todas las aplicaciones tienen esencialmente los mismos objetivos, que son el almacenamiento, distribución y sincronización de ficheros entre varios dispositivos y varios usuarios diferentes, existen diferencias bastante significativas en cuanto a la funcionalidad y utilidad de las diferentes aplicaciones analizadas.

El funcionamiento de todas las aplicaciones en algunos aspectos es bastante similar. En todas ellas, por defecto, la descarga del contenido de los ficheros se realiza en el momento que el usuario solicita abrir el fichero, por lo que en general no se tiene acceso a los ficheros si el dispositivo no está conectado a internet. Este comportamiento tiene como objetivo el minimizar el volumen de transferencia de datos por internet y además ahorrar espacio de almacenamiento en el dispositivo. Algunas aplicaciones permiten marcar algunos archivos para que se sincronicen en el dispositivo de forma que se tenga acceso al fichero sin conexión, lo que se asemeja más al comportamiento de las aplicaciones para ordenadores de sobremesa que típicamente no tienen las restricciones de conexión y almacenamiento de los dispositivos móviles. Además, todas las aplicaciones analizadas se apoyan en aplicaciones externas que estén instaladas en el dispositivo para poder abrir y/o editar la mayoría de los tipos de ficheros.

En cuanto a la funcionalidad ofrecida destaca en primer lugar Dropbox como la aplicación más completa, ya que cuenta con todas las funcionalidades. Además es la única que incorpora la gestión de fotografías en álbumes y una característica fundamental para la utilidad de la aplicación, que es la capacidad de edición de ficheros. No sólo permite la edición de ficheros simples (texto plano), sino que además permite la edición con aplicaciones externas que estén instaladas en el dispositivo y sincroniza los cambios realizados en el fichero con la cuenta del usuario. También es la aplicación que es compatible con la mayor cantidad de plataformas de todas las analizadas y su interfaz de usuario es la más intuitiva de todas. Su punto más débil es la capacidad de almacenamiento, ya que inicialmente la cuenta gratuita ofrece sólo 2GB, pero esto se compensa en parte por el hecho de que puede aumentar hasta los 18GB si se recomienda la aplicación a otras personas.

Dropbox es seguida de cerca por Google Drive, que también cuenta con la mayor parte de las funcionalidades de Dropbox excepto las comentadas anteriormente y la carga automática de fotografías del dispositivo. También destaca por ser la única en permitir la apertura y edición de ficheros en formato Google Docs y la que ofrece unas funciones de ordenación y filtrado de ficheros y directorios en la interfaz gráfica

más completas. Además es la aplicación que mayor capacidad de almacenamiento inicial posee, contando con 15GB en la cuenta gratuita.

En cuanto a las aplicaciones Ubuntu One Files y SkyDrive, ambas son bastante similares y bastante más pobres que las otras dos. Cuentan con bastantes menos funcionalidades y además no permiten la edición de ningún tipo de fichero, con lo cual pierden todas las capacidades de sincronización desde el dispositivo hacia el servidor.

Considerando todo lo anterior podemos concluir que la mejor aplicación de las analizadas es Dropbox, seguida por Google Drive, mientras que las dos aplicaciones restantes quedan muy por debajo.

3. ANÁLISIS DEL SISTEMA

En este capítulo se va a definir cuál es el problema que queremos solucionar con este sistema mediante una descripción general del mismo. Después se va a describir cuál va a ser la funcionalidad del sistema mediante una serie de casos de uso. Por último, a partir de los casos de uso recogidos se va a especificar de qué forma se conseguirá este funcionamiento mediante una serie de requisitos que servirán para el desarrollo posterior del sistema y ayudarán en la toma de decisiones.

3.1.Descripción general

El sistema desarrollado permite compartir, distribuir y sincronizar ficheros entre varios dispositivos diferentes. Esto permite tener los ficheros disponibles en todos los dispositivos que utilice el usuario y sincronizar de forma automática los cambios realizados en los ficheros en todos los dispositivos.

El sistema contará con un servidor que se comunica con las aplicaciones clientes instaladas en los dispositivos móviles de los usuarios. Es en el servidor donde se almacenan los ficheros y a través del cual se sincronizan entre los diferentes dispositivos y usuarios.

Este sistema será utilizado a través de una aplicación cliente para dispositivos móviles con sistema operativo Android, de forma que se deben tener en cuenta las características y limitaciones de este tipo de dispositivos. La interfaz gráfica de usuario está diseñada para que sea usable en estos dispositivos, los cuales tienen una pantalla de tamaño reducido y táctil. Además, se debe tener en cuenta la capacidad limitada de proceso y almacenamiento de estos dispositivos, así como la limitación de su conexión a internet, que se debe optimizar para reducir el tráfico de datos. Por este motivo los ficheros se descargan en el dispositivo móvil a petición del usuario, a diferencia de otras aplicaciones para sincronización de ficheros para ordenador, ya que de esta forma se consigue un ahorro muy importante tanto del uso de la conexión a internet del dispositivo como de su espacio de almacenamiento. Además, el protocolo utilizado intenta igualmente minimizar la cantidad de información intercambiada entre el dispositivo y el servidor, contribuyendo también a la reducción del tráfico.

3.1.1. Navegación por etiquetas

Una de las características diferenciales del sistema desarrollado respecto de otros sistemas similares es el sistema de etiquetas para los ficheros.

Una etiqueta consiste en un metadato de tipo textual asociado a un fichero o conjunto de ficheros que permite su búsqueda dentro del sistema, es decir, permite filtrar los ficheros que se muestran respecto de la etiqueta de forma que se acota el conjunto de ficheros mostrados. Según se van añadiendo más etiquetas al filtro, el conjunto de ficheros se reduce y por tanto la búsqueda se refina.

Este sistema permite al usuario asignar una o varias etiquetas a los ficheros, de forma que se posibilita la clasificación de los ficheros y se facilita la navegación por los mismos, haciendo más sencillo encontrar un determinado fichero o un conjunto de ellos. Para ello se pueden filtrar los ficheros mostrados por una o varias etiquetas simultáneamente. Cabe destacar que el sistema no permite la utilización de directorios. Se podría decir que las etiquetas vienen a sustituir a los directorios que podemos encontrar en otros sistemas similares.

Las etiquetas cuentan con la ventaja de que no requieren una organización jerárquica como los directorios. El sistema jerárquico tiene el inconveniente de que es necesario conocer la ruta del fichero, es decir, el orden en que se aplican los distintos subdirectorios durante la búsqueda del fichero. Esto quiere decir que en un sistema jerárquico sólo existe un camino de búsqueda para llegar a un fichero. Por el contrario, en el caso de un sistema de etiquetas, durante la búsqueda de un fichero, no importa el orden en el que se aplican al filtro las etiquetas, por lo que podemos considerar que existen distintos caminos para llegar al fichero deseado. Por ejemplo, si queremos buscar un fichero que tiene las etiquetas *fotos*, *vacaciones* y *2014*, podemos aplicar las etiquetas al filtro en cualquier orden. En cambio, en un sistema de directorios, para buscar el mismo fichero tendríamos que conocer el orden de los criterios por los que se organizó el sistema de ficheros, es decir, debemos los subdirectorios *fotos*, *vacaciones* y *2014* en el orden adecuado.

3.2.Casos de Uso

En esta sección se va a exponer una serie de casos de uso que nos permitirán identificar los requisitos funcionales del sistema en relación a los roles de usuario. Un caso de uso es una especificación de los pasos que deberán realizarse para llevar a cabo algún proceso. Para especificar un caso de uso es necesario establecer el actor o los actores que intervienen en el mismo. Los actores son los personajes o entidades externas al sistema que interactúan con el mismo. Cada caso de uso describe una secuencia de interacciones típica entre el sistema y sus actores, por lo general entre el sistema y el usuario (describe una forma de usar el sistema por parte del usuario).

3.2.1. Diagrama de Casos de Uso

El diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Especifica la comunicación y el comportamiento del sistema mediante su interacción con los actores (generalmente los usuarios aunque también pueden ser otros sistemas o entidades externas) [14].

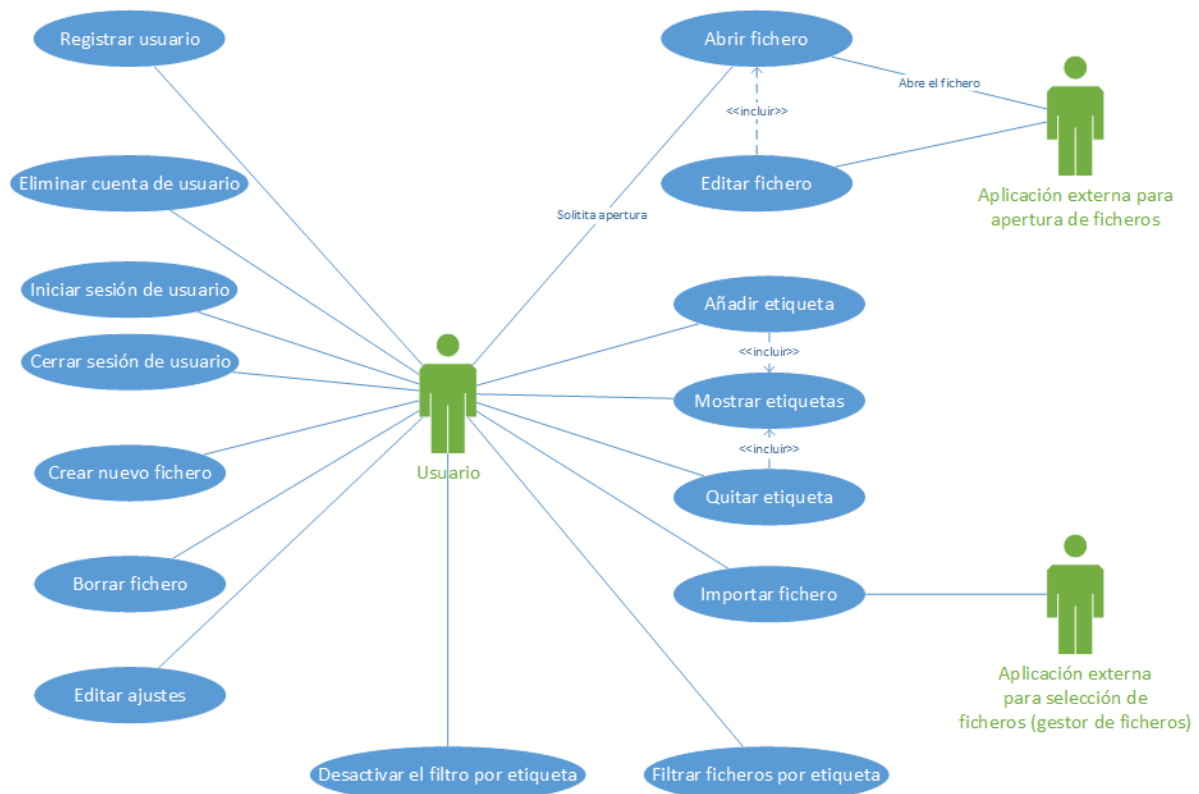


Ilustración 9. Diagrama de casos de uso

3.2.2. Especificación de Casos de Uso

En esta sección se describirá más en detalle cada uno de los casos de uso. Para ello se utilizará una tabla para cada caso de uso que describe sus características.

A continuación se presenta una plantilla de la tabla que se utilizará para describir en detalle cada uno de los casos de uso.

Identificador	Título
Actores	
Objetivo	
Precondiciones	
Postcondiciones	
Escenario Principal	
Escenario Alternativo	

Tabla 1. Plantilla para Casos de Uso

La descripción de cada uno de los campos de la tabla anterior es la siguiente:

- **Identificador:** Permite identificar unívocamente cada caso de uso. Tiene la siguiente estructura: CU-XX. Consta de un prefijo CU que indica que se trata de un caso de uso seguido de un número incremental único para cada caso de uso (XX del esquema anterior).
- **Título:** Es el nombre descriptivo asignado al caso de uso.
- **Actores:** Agentes externos que interactúan con el sistema. Cada actor representa un rol que adopta una entidad externa (típicamente roles de usuario).
- **Objetivo:** Describe la finalidad del caso de uso.
- **Precondiciones:** Condiciones necesarias que deben darse para que el caso de uso pueda realizarse.
- **Postcondiciones:** Efectos y consecuencias en el sistema que provoca la ejecución del caso de uso.
- **Escenario Principal:** Serie de pasos que se producen en la interacción entre los actores y el sistema en el escenario normal del caso de uso. Describen el flujo de acciones principal del caso de uso.

- **Escenario Alternativo:** Secuencias de pasos alternativas respecto al escenario principal. Describen flujos de acciones alternativos respecto del escenario principal que se producen debido a la evaluación de alguna condición de error o algún otro factor.

Una vez descritos cada uno de los campos de la plantilla, se procede a exponer las tablas de los casos de uso.

CU-01	Registrar usuario
Actores	Usuario.
Objetivo	Dar de alta un nuevo usuario en el sistema.
Precondiciones	La aplicación está instalada en el dispositivo.
Postcondiciones	El usuario estará registrado en el sistema.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el icono de la aplicación en el menú de aplicaciones del dispositivo. 2. El sistema comprueba que no hay una sesión de usuario ya iniciada en el dispositivo. 3. El usuario elige la opción de registro de nuevo usuario. 4. El usuario introduce la información del usuario. 5. El usuario pulsa el botón para registrar usuario. 6. El sistema muestra un mensaje indicando que el usuario ha sido registrado correctamente.
Escenario Alternativo	<p>6a. El identificador de usuario introducido ya existe en el sistema:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que el identificador de usuario ya existe en el sistema y solicitando al usuario que seleccione otro identificador. 2. Se vuelve al paso 4 del escenario principal. <p>6b. No hay conexión con el servidor:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no se puede establecer una conexión con el servidor y sugiere al usuario que lo reintente más tarde. 2. Se vuelve al paso 3 del escenario principal. <p>6c. El usuario ha dejado sin rellenar alguno de los campos de información requerida:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando al usuario que rellene todos los campos. 2. Se vuelve al paso 4 del escenario principal.

Tabla 2. CU-01. Registrar usuario

CU-02	Eliminar cuenta de usuario
Actores	Usuario.
Objetivo	Dar de baja a un usuario en el sistema.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.
Postcondiciones	Sesión de usuario cerrada, usuario dado de baja y toda su información y ficheros son eliminados del sistema.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de eliminar cuenta. 2. El sistema muestra un mensaje indicando que la cuenta de usuario ha sido eliminada. Se muestra la pantalla de bienvenida.
Escenario Alternativo	2b. No hay conexión con el servidor: <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no se puede establecer una conexión con el servidor y sugiere al usuario que lo reintente más tarde.

Tabla 3. CU-02. Eliminar cuenta de usuario

CU-03	Iniciar sesión de usuario
Actores	Usuario.
Objetivo	Un usuario inicia sesión en el sistema en este dispositivo introduciendo sus credenciales.
Precondiciones	La aplicación está instalada en el dispositivo.
Postcondiciones	La sesión del usuario estará iniciada en el dispositivo.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el icono de la aplicación en el menú de aplicaciones del dispositivo. 2. El sistema comprueba que no hay una sesión de usuario ya iniciada en el dispositivo. 3. El usuario elige la opción de inicio de sesión. 4. El usuario introduce su identificador y contraseña. 5. El usuario pulsa el botón para iniciar sesión. 6. El sistema muestra un mensaje indicando que la sesión de usuario ha sido iniciada en este dispositivo. 7. Se muestra la página principal de la aplicación.
Escenario Alternativo	6a. El identificador de usuario y la contraseña introducidos no son correctos: <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que el identificador de usuario y la contraseña introducidos no son correctos. 2. Se vuelve al paso 4 del escenario principal. 6b. No hay conexión con el servidor: <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no se puede establecer una conexión con el servidor y sugiere al usuario que lo reintente más tarde. 2. Se vuelve al paso 3 del escenario principal.

Tabla 4. CU-03. Iniciar sesión de usuario

CU-04	Cerrar sesión de usuario
Actores	Usuario.
Objetivo	Un usuario cierra la sesión en este dispositivo.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.
Postcondiciones	Sesión de usuario cerrada en el dispositivo.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de cerrar sesión. 2. El sistema muestra un mensaje indicando que la sesión de usuario ha sido cerrada. 3. Se muestra la pantalla de bienvenida.
Escenario Alternativo	

Tabla 5. CU-04. Cerrar sesión de usuario

CU-05	Crear nuevo fichero
Actores	Usuario.
Objetivo	Crear un nuevo fichero vacío en el sistema.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.
Postcondiciones	Fichero vacío creado en el sistema.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de crear nuevo fichero. 2. El usuario introduce un nombre para el nuevo fichero. 3. El sistema muestra un mensaje indicando que el fichero ha sido creado correctamente.
Escenario Alternativo	<p>3a. Ya existe un fichero con el nombre introducido en el sistema:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que ya existe un fichero con el nombre introducido y solicitando al usuario que introduzca otro nombre. 2. Se vuelve al paso 2 del escenario principal.

Tabla 6. CU-05. Crear nuevo fichero

CU-06	Importar fichero
Actores	Usuario y aplicación externa con la que se selecciona el fichero a importar (gestor de ficheros).
Objetivo	Importar en el sistema un fichero ya existente en el dispositivo.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.
Postcondiciones	Fichero importado en el sistema.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de importar fichero. 2. El sistema abre una aplicación externa que esté instalada

	<p>en el dispositivo que permita seleccionar el fichero (típicamente un gestor de ficheros).</p> <ol style="list-style-type: none"> 3. El usuario selecciona el fichero a importar en la aplicación externa. 4. El usuario introduce un nombre para el nuevo fichero a importar. 5. El sistema muestra un mensaje indicando que el fichero ha sido importado correctamente.
Escenario Alternativo	<p>2a. No hay ninguna aplicación instalada en el dispositivo que permita seleccionar un fichero existente en el dispositivo:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no hay ninguna aplicación instalada en el dispositivo que permita seleccionar el fichero a importar. <p>5a. Ya existe un fichero con el mismo nombre en el sistema:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que ya existe un fichero con el mismo nombre y solicitando al usuario que introduzca otro nombre para el fichero. 2. Se vuelve al paso 4 del escenario principal.

Tabla 7. CU-06. Importar fichero

CU-07	Abrir fichero
Actores	Usuario y aplicación externa con la que se abre el fichero.
Objetivo	Abrir un fichero existente en el sistema.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.
Postcondiciones	Fichero abierto en una aplicación externa.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa un fichero de la lista de ficheros. 2. El sistema comprueba que el fichero está obsoleto o no está presente físicamente en el dispositivo y que hay conexión al servidor. 3. El sistema descarga el fichero del servidor. 4. El sistema abre el fichero en una aplicación externa que esté instalada en el dispositivo dependiendo del tipo de fichero.
Escenario Alternativo	<p>2a. No hay conexión con el servidor y el fichero no está físicamente en el dispositivo:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no se puede descargar el fichero ya que no hay conexión con el servidor. <p>2b. El fichero si está presente físicamente en el dispositivo y sincronizado, ha sido modificado localmente o ha sido creado localmente y aún no sincronizado en el servidor:</p> <ol style="list-style-type: none"> 1. Se va al paso 4 del escenario principal. <p>2c. No hay conexión con el servidor y el fichero si está</p>

	<p>presente físicamente en el dispositivo pero está obsoleto:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que el fichero que se va a abrir está obsoleto debido a que no se puede descargar el fichero actualizado porque no hay conexión con el servidor y se pregunta al usuario si quiere reintentar la descarga, abrir el fichero obsoleto o cancelar. 2. Si el usuario elige reintentar se va al paso 2 del escenario principal. Si el usuario elige abrir el fichero obsoleto se va al paso 4 del escenario principal. <p>2d. Existe un conflicto de versiones del fichero:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando al usuario que existe un conflicto de versiones y se le pregunta si desea mantener la versión del servidor, la versión del dispositivo o cancelar. 2. Si el usuario elige mantener la versión del servidor se borra la versión del dispositivo, se desmarca el fichero como con conflicto y se va al paso 1 del escenario principal. Si el usuario elige mantener la versión del dispositivo se desmarca el fichero como con conflicto, se marca para que el cambio sea actualizado en el servidor t se va al paso 1 del escenario principal. <p>4a. No hay ninguna aplicación instalada en el dispositivo que pueda abrir ese tipo de fichero:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no hay ninguna aplicación instalada en el dispositivo que pueda abrir ese tipo de fichero.
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 8. CU-07. Abrir fichero

CU-08	Editar fichero
Actores	Aplicación externa con la que se ha abierto el fichero.
Objetivo	Editar un fichero que se ha abierto con anterioridad.
Precondiciones	Sesión de usuario iniciada, servicio de la aplicación iniciado en el dispositivo y fichero abierto en aplicación externa.
Postcondiciones	Cambios del fichero sincronizados en el sistema.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario edita un fichero abierto mediante una aplicación externa. 2. La aplicación externa guarda los cambios realizados en el fichero. 3. El sistema detecta que se ha escrito en el fichero. 4. El sistema calcula los nuevos metadatos del fichero, como el hash y el tamaño. 5. El sistema confirma que el fichero efectivamente ha cambiado (nuevo hash y/o tamaño no coinciden con los

	antiguos). 6. El sistema marca el fichero como modificado.
Escenario Alternativo	5a. El contenido del fichero no ha sido modificado: <ol style="list-style-type: none"> 1. El sistema determina que realmente el contenido del fichero no ha sido modificado (nuevo hash y tamaño coinciden con los antiguos). 2. El sistema no marca el fichero como modificado. 6b. El contenido del fichero ha sido modificado y éste estaba obsoleto: <ol style="list-style-type: none"> 1. El sistema marca el fichero indicando que existe conflicto de versiones.

Tabla 9. CU-08. Editar fichero

CU-09	Borrar fichero
Actores	Usuario.
Objetivo	Borrar del sistema un fichero.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.
Postcondiciones	Fichero eliminado del sistema.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario abre el menú contextual de un fichero. 2. El usuario pulsa el botón de borrar fichero. 3. El sistema muestra un mensaje indicando que el fichero ha sido borrado correctamente.
Escenario Alternativo	

Tabla 10. CU-09. Borrar fichero

CU-10	Mostrar etiquetas
Actores	Usuario.
Objetivo	Mostrar las etiquetas que tiene un fichero que se encuentre en el sistema.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.
Postcondiciones	Se muestra la pantalla de etiquetas del fichero.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario abre el menú contextual de un fichero. 2. El usuario pulsa el botón para mostrar las etiquetas. 3. El sistema muestra una pantalla con las etiquetas del fichero.
Escenario Alternativo	

Tabla 11. CU-10. Mostrar etiquetas

CU-11	Añadir etiqueta
Actores	Usuario.
Objetivo	Añadir una etiqueta a un fichero que se encuentre en el sistema.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla de etiquetas del fichero.
Postcondiciones	Etiqueta añadida al fichero.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de añadir etiqueta. 2. El usuario introduce el nombre de la etiqueta que quiere añadir. 3. El sistema muestra un mensaje indicando que la etiqueta ha sido añadida al fichero correctamente.
Escenario Alternativo	3a. El fichero ya tiene la etiqueta introducida: <ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que el fichero ya tiene la etiqueta introducida. 2. Se vuelve al paso 2 del escenario principal.

Tabla 12. CU-11. Añadir etiqueta

CU-12	Quitar etiqueta
Actores	Usuario.
Objetivo	Quitar una etiqueta a un fichero que se encuentre en el sistema.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla de etiquetas del fichero.
Postcondiciones	Etiqueta quitada del fichero.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario abre el menú contextual de una etiqueta. 2. El usuario pulsa el botón de quitar etiqueta. 3. El sistema muestra un mensaje indicando que la etiqueta ha sido quitada del fichero correctamente.
Escenario Alternativo	

Tabla 13. CU-12. Quitar etiqueta

CU-13	Filtrar ficheros por etiqueta
Actores	Usuario.
Objetivo	Filtrar los ficheros que se muestran en la lista de ficheros en pantalla de forma que sólo se muestren los ficheros que tengan una determinada etiqueta. Además, sólo se mostrarán en la lista de etiquetas las que pertenezcan a los ficheros mostrados en la lista de ficheros filtrada.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.

Postcondiciones	Sólo se muestran los ficheros con la etiqueta seleccionada y las etiquetas de los mismos.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario selecciona una etiqueta de las que se muestran en la pantalla. 2. El sistema muestra la lista de ficheros que contienen esa etiqueta y sus etiquetas.
Escenario Alternativo	

Tabla 14. CU-13. Filtrar ficheros por etiqueta

CU-14	Desactivar el filtro por etiqueta
Actores	Usuario.
Objetivo	Desactivar el filtrado por etiquetas de los ficheros. Se mostrarán todos los ficheros existentes en el sistema así como todas las etiquetas que estos tengan.
Precondiciones	Sesión de usuario iniciada, sistema mostrando la pantalla principal y filtrado por etiquetas activado.
Postcondiciones	Se mostrarán todos los ficheros que hay en el sistema y todas las etiquetas de los mismos.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de mostrar todos los ficheros. 2. El sistema muestra la lista de todos los ficheros y todas sus etiquetas.
Escenario Alternativo	

Tabla 15. CU-14. Desactivar el filtro por etiqueta

CU-15	Editar ajustes
Actores	Usuario.
Objetivo	Editar los ajustes del sistema en el dispositivo.
Precondiciones	Sesión de usuario iniciada y sistema mostrando la pantalla principal.
Postcondiciones	Ajustes modificados.
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ajustes. 2. El sistema muestra la pantalla de ajustes. 3. El usuario pulsa en el ajuste que desea editar. 4. El sistema muestra una pantalla que permite modificar el valor del ajuste. 5. El usuario introduce el nuevo valor del ajuste. 6. El usuario pulsa el botón de aceptar. 7. El sistema muestra de nuevo la pantalla de ajustes. 8. El usuario pulsa el botón atrás. 9. El sistema muestra la pantalla principal.
Escenario Alternativo	7a. El nuevo valor del ajuste introducido no es válido:

	<ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que el valor del ajuste introducido no es válido. 2. Se vuelve al paso 5 del escenario principal.
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 16. CU-15. Editar ajustes

3.3.Requisitos del sistema

En esta sección se van a definir los requisitos del sistema. Estos requisitos se van a organizar en las categorías siguientes:

- Requisitos Funcionales
- Requisitos No Funcionales
- Requisitos de Restricción

A continuación se presenta una plantilla de la tabla que se utilizará para describir en detalle cada uno de los requisitos.

Identificador	Título		
Fuente		Necesidad	
Prioridad		Estabilidad	
Descripción			

Tabla 17. Plantilla para requisitos

La descripción de cada uno de los campos de la tabla anterior es la siguiente:

- **Identificador:** Permite identificar unívocamente cada requisito. Tiene la siguiente estructura: RX-YY. Consta de un prefijo que indica que se trata de un requisito (sería la R del esquema anterior) seguido de un indicador de la categoría del requisito (sería la X del esquema anterior). Los valores de este prefijo son:
 - RF: para los requisitos funcionales.
 - RNF: para los requisitos no funcionales.
 - RR: para requisitos de restricción.

Por último incluye un número incremental único para cada requisito de ese tipo (YY del esquema anterior).

- **Título:** Es el nombre descriptivo asignado al requisito.
- **Fuente:** Indica la procedencia del requisito.

- **Prioridad:** Indica la prioridad de cumplimiento del requisito. Puede ayudar a decidir la planificación del desarrollo. Puede tomar los valores alta, media y baja.
- **Necesidad:** Indican la necesidad del cumplimiento de cada requisito. Los posibles valores que puede tomar son: esencial, deseable y opcional. Los requisitos esenciales se deben cumplir de forma obligatoria pero los demás pueden ser negociables.
- **Estabilidad:** Indican la estabilidad del requisito a lo largo del desarrollo del sistema, es decir, indican si el requisito va a permanecer igual durante todo el desarrollo o si, al contrario, puede cambiar en algún momento y por tanto merecen ser revisados durante el desarrollo. Los posibles valores que puede tomar son: estable e inestable.

A continuación se presenta cada uno de los requisitos especificados mediante tablas.

3.3.1. Requisitos funcionales

RF-01 Registro de usuario			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que no estén ya registrados podrán registrarse en el sistema aportando como mínimo un identificador de usuario y una contraseña.		

Tabla 18. RF-01. Registro de usuario

RF-02 Eliminación de cuenta de usuario			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que estén ya registrados podrán dar de baja su cuenta y eliminar toda su información y ficheros del sistema.		

Tabla 19. RF-02. Eliminación de cuenta de usuario

RF-03 Inicio de sesión de usuario			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios registrados en el sistema podrán iniciar sesión en el dispositivo aportando su identificador de usuario y su contraseña.		

Tabla 20. RF-03. Inicio de sesión de usuario

RF-04 Cierre de sesión de usuario			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán cerrar su sesión en el dispositivo.		

Tabla 21. RF-04. Cierre de sesión de usuario

RF-05 Creación de nuevo fichero			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán crear ficheros nuevos en el sistema introduciendo el nombre para el nuevo fichero.		

Tabla 22. RF-05. Creación de nuevo fichero

RF-06 Importación de fichero			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán importar ficheros ya existentes en el dispositivo.		

Tabla 23. RF-06. Importación de fichero

RF-07 Apertura de fichero			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán abrir los ficheros que se encuentren en el sistema.		

Tabla 24. RF-07. Apertura de fichero

RF-08 Edición de fichero			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán editar los ficheros que se encuentren en el sistema.		

Tabla 25. RF-08. Edición de fichero

RF-09 Borrado de fichero			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán borrar los ficheros que se encuentren en el sistema.		

Tabla 26. RF-09. Borrado de fichero

RF-10 Visualización de la lista de ficheros			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán visualizar la lista de todos los ficheros existentes en el sistema.		

Tabla 27. RF-10. Visualización de la lista de ficheros

RF-11 Visualización de etiquetas			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán visualizar las etiquetas que posee un fichero que se encuentre en el sistema.		

Tabla 28. RF-11. Visualización de etiquetas

RF-12 Adición de etiqueta			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán añadir etiquetas a los ficheros que se encuentren en el sistema.		

Tabla 29. RF-12. Adición de etiqueta

RF-13 Eliminación de etiqueta			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán eliminar las etiquetas que posean los ficheros que se encuentren en el sistema.		

Tabla 30. RF-13. Eliminación de etiqueta

RF-14 Filtrado de ficheros por etiqueta			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán filtrar los ficheros que se muestran basándose en que éstos posean una determinada etiqueta. Es decir, los usuarios pueden añadir etiquetas al filtro para que se muestren sólo los ficheros que contienen todas las etiquetas del filtro.		

Tabla 31. RF-14. Filtrado de ficheros por etiqueta

RF-15 Eliminación de etiqueta del filtro			
Fuente	Autor del proyecto	Necesidad	Opcional
Prioridad	Baja	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán eliminar una etiqueta del filtro de forma que se muestren sólo los ficheros que posean las etiquetas del filtro. Es la operación inversa al filtrado de ficheros por etiqueta especificada en el requisito anterior.		

Tabla 32. RF-15. Filtrado de ficheros por etiqueta

RF-16 Desactivación del filtrado por etiqueta			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán desactivar el filtrado de ficheros por etiquetas de forma que se muestren todos los ficheros.		

Tabla 33. RF-16. Desactivación del filtrado por etiqueta

RF-17 Edición de ajustes			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los usuarios que hayan iniciado sesión podrán editar los ajustes del sistema.		

Tabla 34. RF-17. Edición de ajustes

3.3.2. Requisitos no funcionales

RNF-01 Descarga de ficheros a petición del usuario			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Los ficheros se descargarán del servidor (en caso de no encontrarse ya en el dispositivo) en el momento en que el usuario solicite su apertura, con lo que se ahorra espacio de almacenamiento en el dispositivo y se reduce el tráfico de datos.		

Tabla 35. RNF-01. Descarga de ficheros a petición del usuario

RNF-02 Sincronización asíncrona			
Fuente	Tutor del proyecto	Necesidad	Deseable
Prioridad	Media	Estabilidad	Estable
Descripción	La sincronización con el servidor se realizará de forma asíncrona, de forma que no afecte al funcionamiento ni capacidad de respuesta de la interfaz de usuario.		

Tabla 36. RNF-02. Sincronización asíncrona

RNF-03 Comprobación de modificación de ficheros			
Fuente	Autor del proyecto	Necesidad	Opcional
Prioridad	Baja	Estabilidad	Estable
Descripción	<p>Cuando se detecte que un fichero ha sido guardado por una aplicación externa, antes de proceder a marcar el fichero como modificado en los metadatos y sincronizar los cambios, se comprueba si realmente ha sido modificado. Para ello se realizan los siguientes pasos:</p> <ol style="list-style-type: none">1. Primero se comprueba si el tamaño del fichero o la fecha de última modificación del fichero proporcionada por el sistema de ficheros ha cambiado. Si no han cambiado sabemos que el fichero no ha sido realmente modificado y el algoritmo finaliza.2. Se calcula de nuevo el hash del fichero y se comprueba si ha cambiado. Si el hash y el tamaño del fichero no han cambiado sabemos que el fichero no ha cambiado y el algoritmo finaliza.3. Si el hash o el tamaño del fichero han cambiado sabemos que el fichero ha cambiado realmente ya que su contenido es distinto. <p>Todo este sistema se usa para evitar recalcular el hash en caso de no ser necesario, puesto que es una operación computacionalmente bastante costosa. Se intenta comprobar que el fichero no ha cambiado por otros medios menos costosos (tamaño y fecha de última modificación).</p>		

Tabla 37. RNF-03. Comprobación de modificación de ficheros

RNF-04 Control de versiones y conflictos de fichero			
Fuente	Autor del proyecto	Necesidad	Deseable
Prioridad	Media	Estabilidad	Estable
Descripción	Existirá un sistema de control de versiones de fichero que permita asegurar que cuando se modifica un fichero siempre se haga sobre la versión más reciente del mismo. Este sistema detectará los conflictos de versión de fichero y se lo mostrará al usuario para que éste decida con qué versión decide quedarse.		

Tabla 38. RNF-04. Control de versiones y conflictos de fichero

RNF-05 Sincronización de lista de ficheros parcial			
Fuente	Autor del proyecto	Necesidad	Opcional
Prioridad	Baja	Estabilidad	Estable
Descripción	Cuando se produzca la sincronización de la lista de ficheros, es decir, cuando se envíe la lista de los metadatos del servidor al cliente, siempre que sea posible, se enviarán únicamente los metadatos de los ficheros que han sido modificados. Esto significa que siempre que sea posible, se van a enviar únicamente las actualizaciones que se han producido en los ficheros en lugar de enviar la lista de ficheros completa. Esto permitirá reducir el tráfico de datos necesario.		

Tabla 39. RNF-05. Sincronización de lista de ficheros parcial

3.3.3. Requisitos de restricción

RR-01 Conexión a internet			
Fuente	Tutor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	El dispositivo cliente tendrá que contar con conexión a internet para poder comunicarse con el servidor, si bien es cierto que no es necesario que esta conexión sea permanente ya que la aplicación cliente puede funcionar sin conexión y sincronizar los cambios cuando la conexión esté disponible (excepto para el registro de usuario y el inicio de sesión que es necesaria la conexión en el momento que el usuario las solicita). Obviamente el servidor también tendrá que contar con conexión a internet.		

Tabla 40. RR-01. Conexión a internet

RR-02 Gestor de ficheros			
Fuente	Autor del proyecto	Necesidad	Deseable
Prioridad	Media	Estabilidad	Estable
Descripción	Será necesario tener instalado en el dispositivo un gestor de ficheros u otra aplicación que permita seleccionar un fichero existente en el dispositivo para poder usar la funcionalidad de importar archivos.		

Tabla 41. RR-02. Gestor de ficheros

RR-03 Aplicaciones externas para apertura de ficheros			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	Será necesario tener instalada en el dispositivo una aplicación que permita abrir el tipo de fichero que se desea abrir.		

Tabla 42. RR-03. Aplicaciones externas para apertura de ficheros

RR-04 Versión de Android			
Fuente	Autor del proyecto	Necesidad	Esencial
Prioridad	Alta	Estabilidad	Estable
Descripción	La aplicación cliente debe funcionar en el sistema operativo Android. Debe funcionar en la versión 2.2 (Froyo), equivalente al nivel de API 8, o superior.		

Tabla 43. RR-04. Versión de Android

4. DISEÑO DEL SISTEMA

En este capítulo se va a describir el diseño del sistema. En primer lugar se definirá el diseño de una forma más general mediante la descripción de su arquitectura y después se especificará de una forma más detallada mediante los diagramas de clases y de secuencia.

4.1.Arquitectura del sistema

En esta sección se describe el diseño de la arquitectura del sistema.

Para describir la arquitectura de este sistema empezaremos por especificar las diferentes categorías de datos que entran en juego en el sistema:

1. **Ficheros de usuario:** Como se trata de un sistema de almacenamiento, distribución y sincronización de ficheros, tenemos obviamente en el sistema estos ficheros de usuario que van a ser los datos principales del sistema y que son la razón de ser del mismo. Son los que constituyen realmente los datos del sistema (podemos considerar el resto de información como metadatos).
2. **Metadatos:**
 - a. **Metadatos de fichero:** Estos son los metadatos que constituyen información útil relacionada con los ficheros de un usuario. Dentro de los metadatos de ficheros podemos encontrar dos tipos:
 - i. **Metadatos de fichero principales:** Estos metadatos son los metadatos que se sincronizan entre cliente y servidor y por tanto son iguales en ambos.
 - ii. **Metadatos de fichero internos:** Estos son los metadatos de los ficheros que no se sincronizan entre cliente y servidor puesto que sirven únicamente para el correcto funcionamiento interno de los mismos. Son diferentes en el cliente y el servidor (el cliente necesita más metadatos de este tipo para mantener su estado).
 - b. **Metadatos de usuario:** Estos serán los datos sobre cada usuario que maneja el sistema (identificador y contraseña).

La arquitectura del sistema se basa principalmente en dos patrones: Modelo en capas y Cliente-Servidor. Podemos destacar que la arquitectura de este sistema, aunque se basa en estos patrones de diseño conocidos, es bastante particular.

4.1.1. Arquitectura general

De una forma muy general, el sistema sigue un modelo cliente-servidor. El cliente realiza peticiones al servidor y este último se encarga de procesarlas y responderlas, como en cualquier sistema cliente-servidor como se puede observar en la siguiente ilustración.

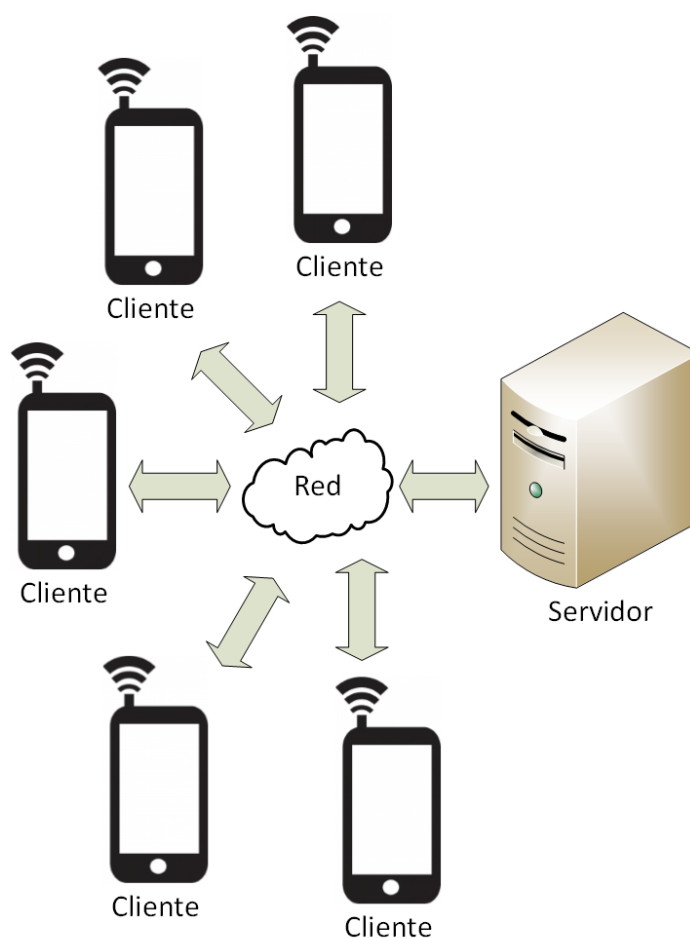


Ilustración 10. Arquitectura general del sistema

A diferencia de lo que ocurre en la mayor parte de los casos, en que el cliente no puede funcionar sin conexión al servidor, este sistema tiene la particularidad de que podemos considerar al cliente y al servidor como dos subsistemas hasta cierto punto autónomos. En este sistema el cliente puede funcionar sin estar conectado al servidor y ofrecer la mayor parte de sus funcionalidades al usuario.

En cuanto a la arquitectura interna de cada uno de estos subsistemas, es diferente en ambos. A continuación pasaremos a describir cada uno de ellos.

4.1.2. Arquitectura del servidor

Podemos considerar que el servidor es un subsistema que tiene una arquitectura en dos capas. La primera es la capa de datos y la segunda sería la lógica. No posee capa de vista ya que no proporciona ninguna forma de interacción con los usuarios (toda la interacción con los usuarios se realiza a través del cliente) ni con otros sistemas.

La capa de datos se divide en dos partes. Por una parte tenemos la gestión de los datos (los ficheros de usuario) y por otra parte tenemos la gestión de los metadatos (metadatos de ficheros y de usuario), esta última muy ligada a la capa lógica. Por tanto podemos concluir que esta capa se comunica únicamente con la capa lógica y con el sistema de almacenamiento.

En cuanto a la capa lógica, ésta se encarga de recibir las peticiones del cliente, procesarlas, realizar las solicitudes necesarias a la capa de datos y generar y enviar las respuestas al cliente. Constituye el núcleo del servidor y realiza la mayor parte de trabajo. Esta capa se comunica con la capa de datos y con el cliente.

4.1.3. Arquitectura del cliente

Podemos considerar que el cliente posee una arquitectura interna en tres capas. La primera capa sería la capa de presentación, la segunda la capa lógica y la tercera la capa de datos.

La capa de presentación es la que se encarga de todos los aspectos relacionados con la interfaz de usuario, es decir, gestionar toda la interacción con el usuario. Básicamente su misión consiste en recibir las peticiones del usuario, que se realizan a través de interacciones como pulsar botones u otros elementos de la interfaz, realizar las solicitudes necesarias a la capa lógica y mostrar al usuario los datos e informaciones requeridos. También se encarga de la interacción con las aplicaciones externas, consistente en solicitarles determinadas funcionalidades y la recibir las respuestas. Por tanto esta capa se comunica con el usuario, con la capa lógica y con las aplicaciones externas.

La capa lógica es la que se encarga de generar y enviar las peticiones al servidor, recibir y procesar las respuestas, realizar las solicitudes necesarias a la capa de datos y realizar las solicitudes necesarias a la capa de presentación. Por tanto esta capa se comunica con la capa de presentación, con el servidor y con la capa de datos.

La capa de datos es similar a la capa de datos del servidor y también se divide en dos partes. Por una parte tenemos la gestión de los datos (los ficheros de usuario) y por otra parte tenemos la gestión de los metadatos (metadatos de ficheros y de usuario), esta última muy ligada a la capa lógica. Por tanto podemos concluir que esta capa se comunica únicamente con la capa lógica y con el sistema de almacenamiento.

La existencia de las tres capas es bastante inusual en las aplicaciones cliente, que por lo general incluyen únicamente la capa de presentación o ésta y la capa lógica (o al menos parte de ella). Por lo general no se incluye la capa de datos ya que normalmente ésta se encuentra únicamente en el servidor.

En este sistema, la existencia de la capa de datos en el cliente es la que permite al cliente funcionar de forma autónoma sin conexión al servidor. Esto implica que el cliente debe mantener ciertos datos localmente para poder funcionar sin servidor y guardar los cambios realizados localmente de forma que se sincronicen en el servidor cuando la conexión con el mismo esté disponible.

Cabe destacar que cuando el cliente funciona sin conexión al servidor, en determinadas circunstancias no podrá proporcionar determinadas funcionalidades, por ejemplo, la apertura de ficheros que no estén disponibles localmente. Esto se debe a que uno de los requisitos del sistema es que la descarga de ficheros sea a petición del usuario, para ahorrar espacio de almacenamiento en el dispositivo y reducir el tráfico de datos. Evidentemente si no hay conexión al servidor tampoco se podrá contar con los cambios que realice el usuario en otros dispositivos ni se podrán sincronizar los cambios realizados localmente en el servidor.

4.2. Diagramas de clases

En esta sección se muestran los diagramas de clases [15] del sistema que describen el sistema desarrollado con mayor detalle. Como se ha explicado anteriormente el sistema consta de dos subsistemas principales que son el cliente y el servidor. Como gran parte de la funcionalidad de ambos subsistemas es la misma, se ha optado por empaquetar todas las clases que proporcionan esta funcionalidad común en un paquete separado a modo de librería. De esta forma se posibilita la reutilización de todo este código de la librería en el cliente y el servidor. Estas clases comunes son principalmente las relacionadas con el protocolo de comunicación, los metadatos comunes y otras funcionalidades que son comunes a cliente y servidor.

A continuación se muestra un diagrama con los paquetes que integran el sistema. En la columna de la izquierda se muestran los paquetes que forman parte del cliente, en la columna central los que forman parte de la librería común y en la columna de la derecha los que forman parte del servidor.

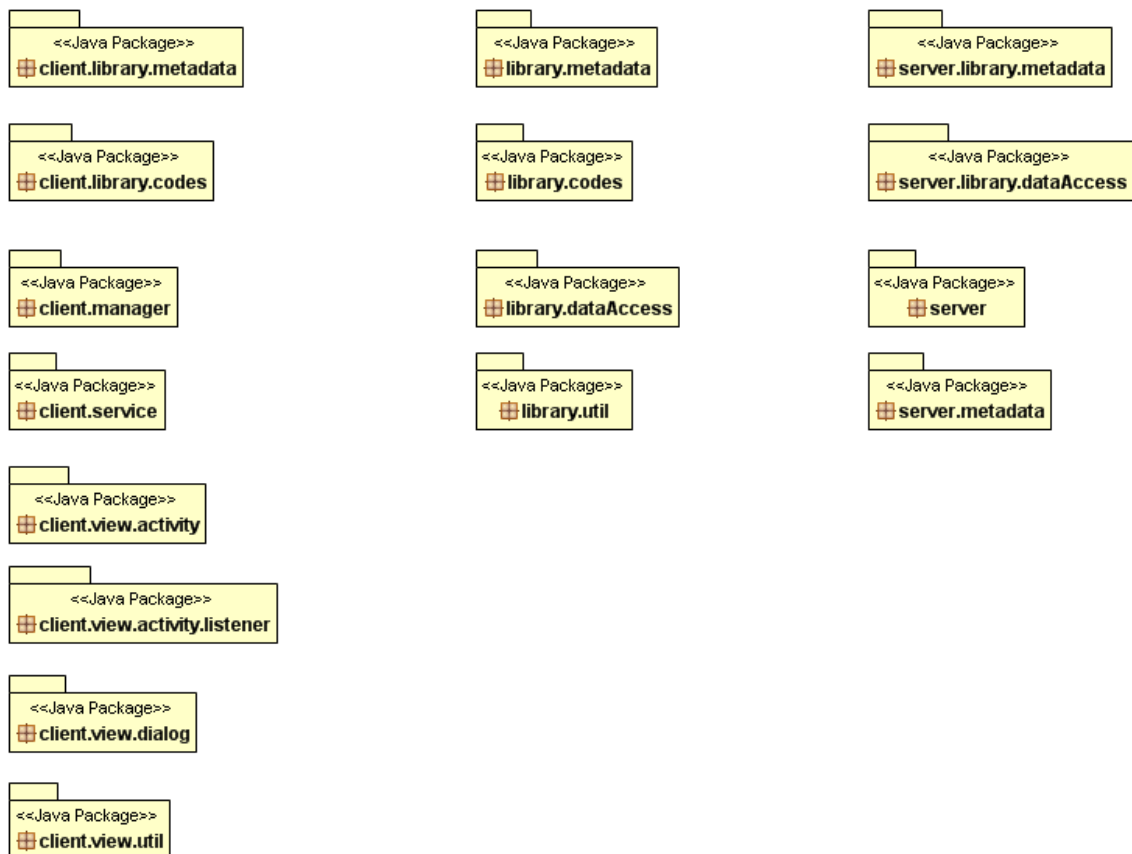


Ilustración 11. Diagrama de paquetes general

A continuación se incluyen a continuación los diagramas de clases de cada una de las partes del sistema descritas anteriormente. Para que los diagramas sean más legibles y sencillos de comprender no se muestran ni los atributos ni los métodos de las clases. Únicamente se muestra el nombre de cada clase, el paquete al que pertenece, un icono indicativo del tipo de clase (si se trata de una clase se indica con una C verde, si es una interfaz con una I morada y si es un enumerado se indica con una E marrón) y las relaciones entre las clases.

En primer lugar se muestra el diagrama de clases de la librería común. Hay que tener en cuenta que todas las clases de la librería entran en funcionamiento tanto en el cliente como en el servidor.

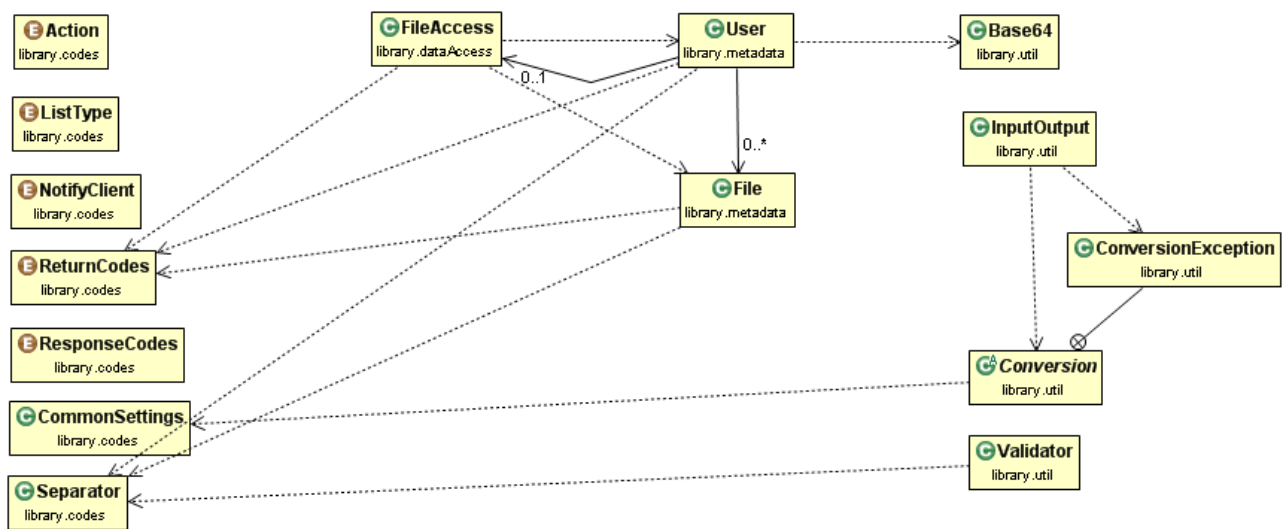


Ilustración 12. Diagrama de clases de la librería compartida

A continuación se muestra el diagrama de clases del cliente del sistema. En el caso de este diagrama se han omitido las relaciones de dependencia entre clases para simplificar el diagrama y hacerlo más legible. Se ha omitido este tipo de relación debido a que es la relación más débil entre clases, ya que indica que una clase utiliza a otra en algún momento específico, sea porque se recibe como parámetro o se utiliza como variable local. Todo esto significa que no se trata de una relación estructural entre clases, como ocurre con otras relaciones, por lo que tampoco aporta una información tan valiosa.

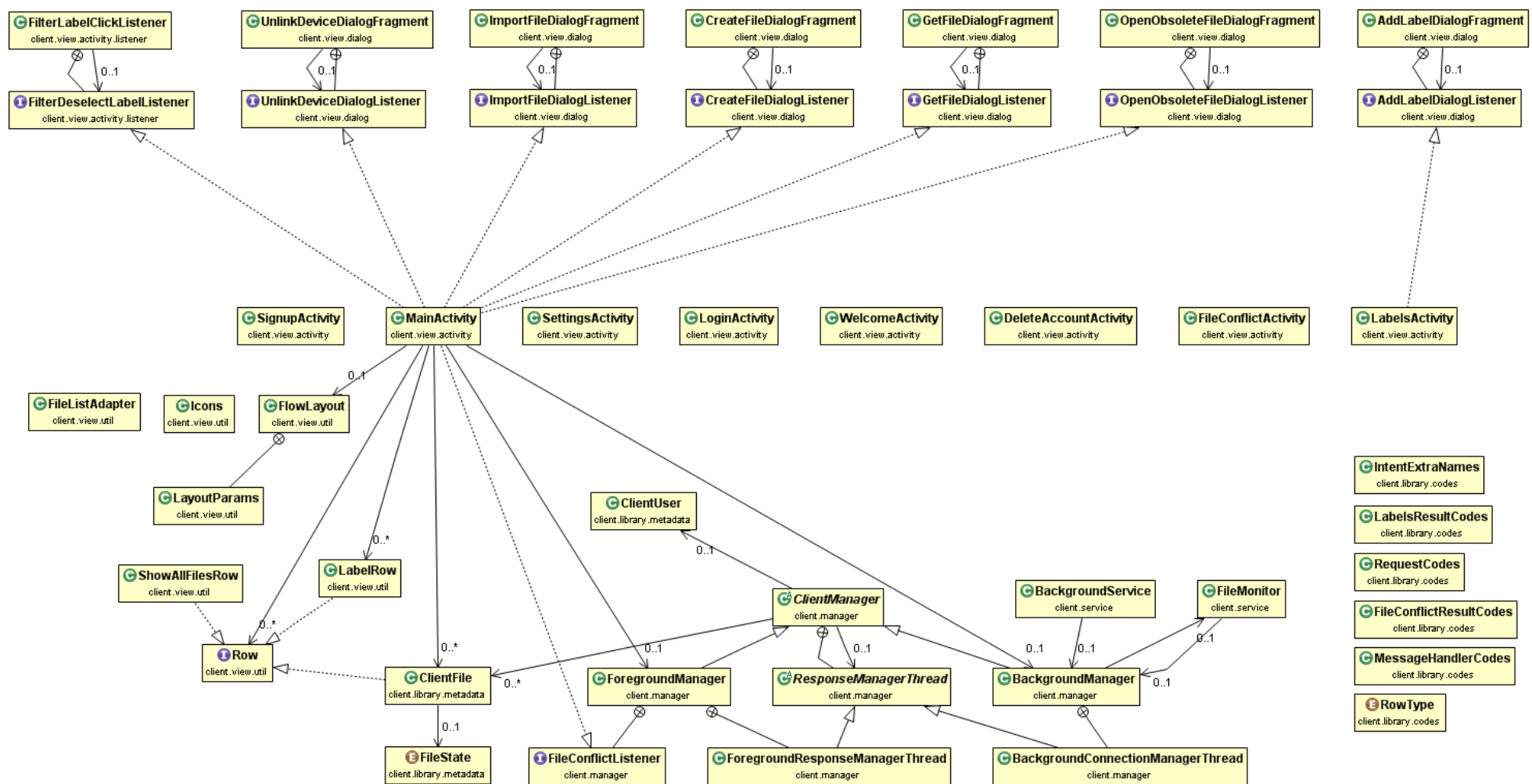


Ilustración 13. Diagrama de clases del cliente

Por último se muestra el diagrama de clases del servidor. Es el diagrama más sencillo y el que cuenta con menor número de clases ya que el servidor es sensiblemente menos complejo que el cliente. Aun así, hay que tener en cuenta que todas las clases de la librería compartida también formarán parte del servidor como se ha explicado anteriormente.

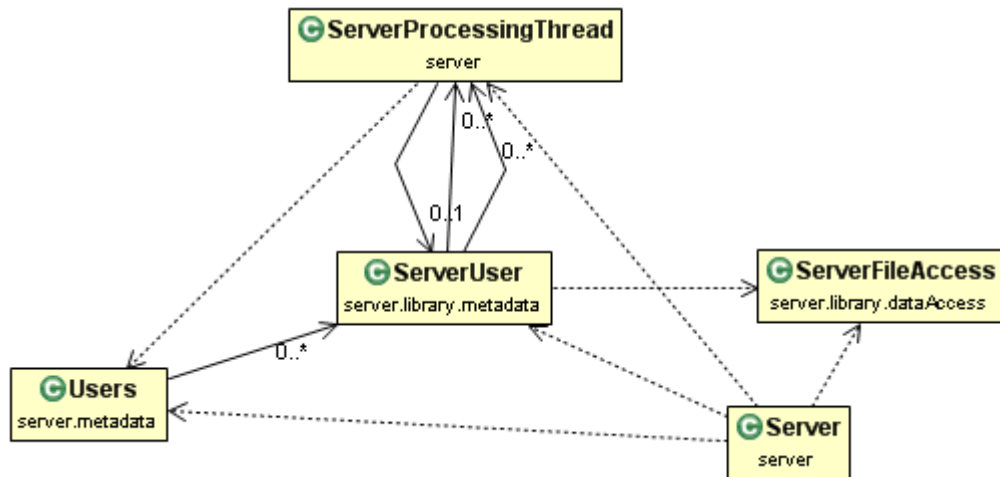


Ilustración 14. Diagrama de clases del servidor

4.3. Diagramas de secuencia

En esta sección se presentan los diagramas de secuencia correspondientes a algunos de los casos de uso especificados en este documento. Se incluyen únicamente los diagramas de secuencia que se consideran más relevantes para poder entender el funcionamiento del sistema. Estos diagramas de secuencia describen las interacciones que se producen entre los distintos objetos o componentes del sistema a lo largo del tiempo y cada uno de ellos modela las interacciones que se producen durante el desarrollo de un determinado caso de uso [16] [17].

Para que los diagramas resulten más sencillos, y siguiendo con la descripción realizada en los casos de uso, en la mayoría de ellos se considera que el usuario ya tiene la aplicación cliente abierta y que se encuentra en la pantalla adecuada para poder realizar la acción requerida, es decir, en la pantalla donde se encuentra el botón o elemento de la interfaz gráfica que le permite iniciar la acción a realizar. Además, los diagramas están enormemente simplificados, de forma que se incluyen sólo los objetos o componentes principales del sistema, así como las interacciones principales, aunque en realidad intervengan muchos más. Esta simplificación resulta necesaria dado que sería muy complicado representar todas las interacciones, ya que resultarían diagramas demasiado extensos y complejos.

4.3.1. Registrar usuario

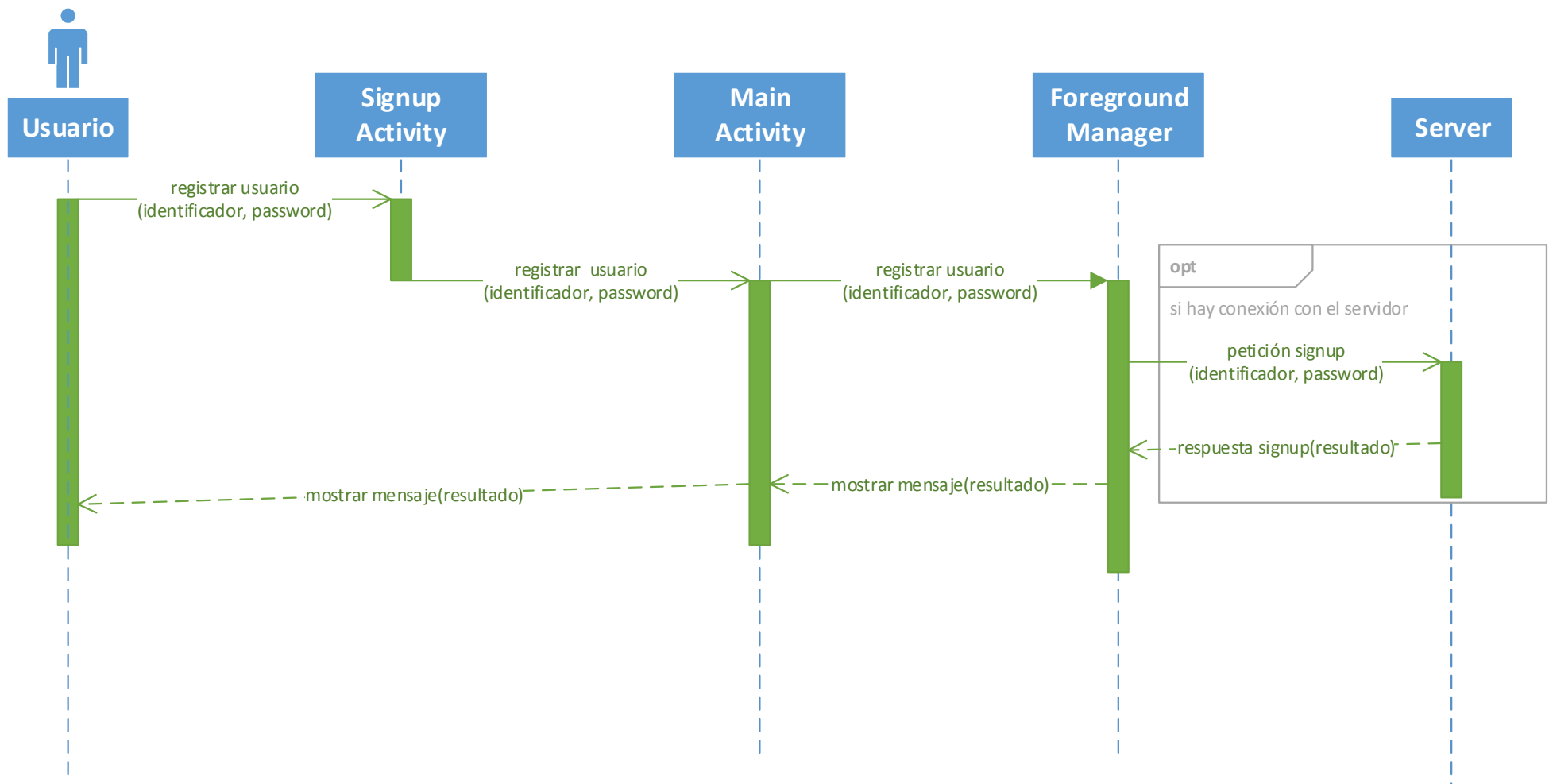


Ilustración 15. Diagrama de secuencia. Registrar usuario

4.3.2. Iniciar sesión de usuario

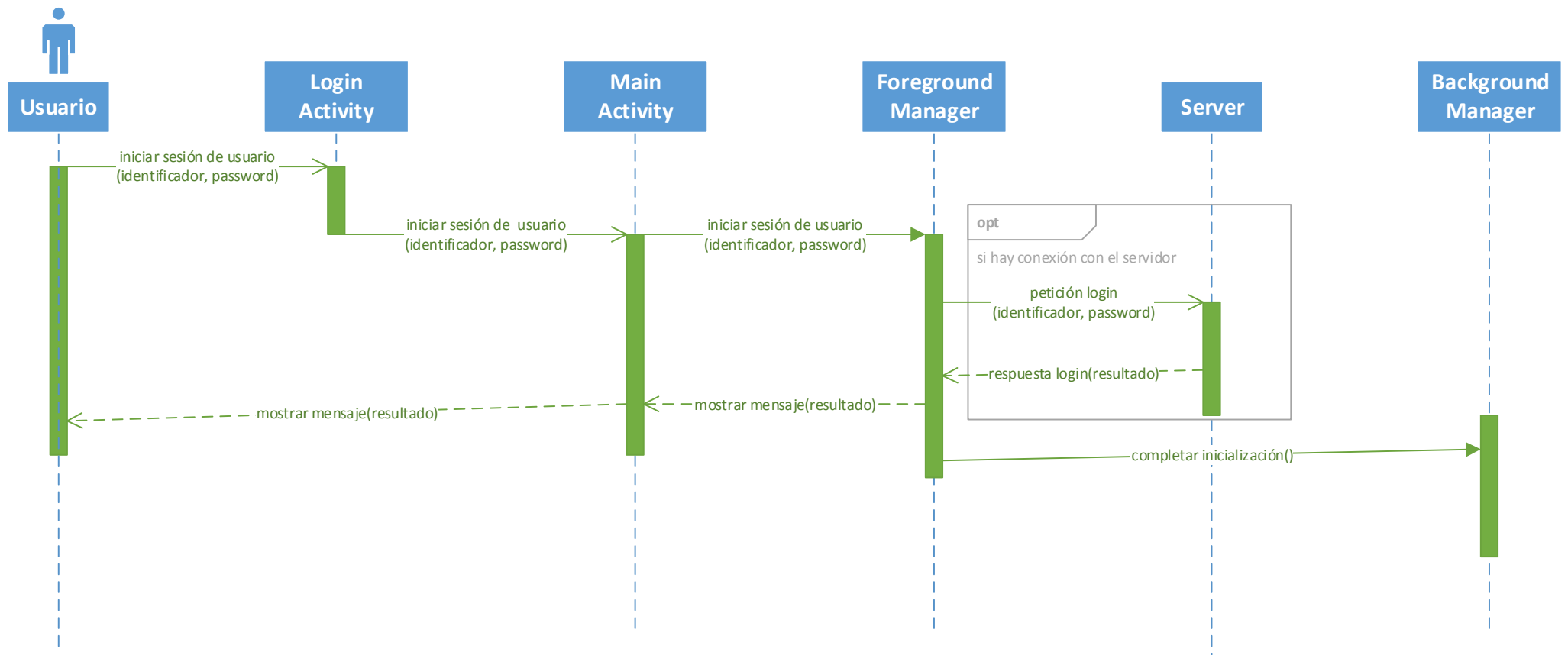


Ilustración 16. Diagrama de secuencia. Iniciar sesión de usuario

4.3.3. Importar fichero

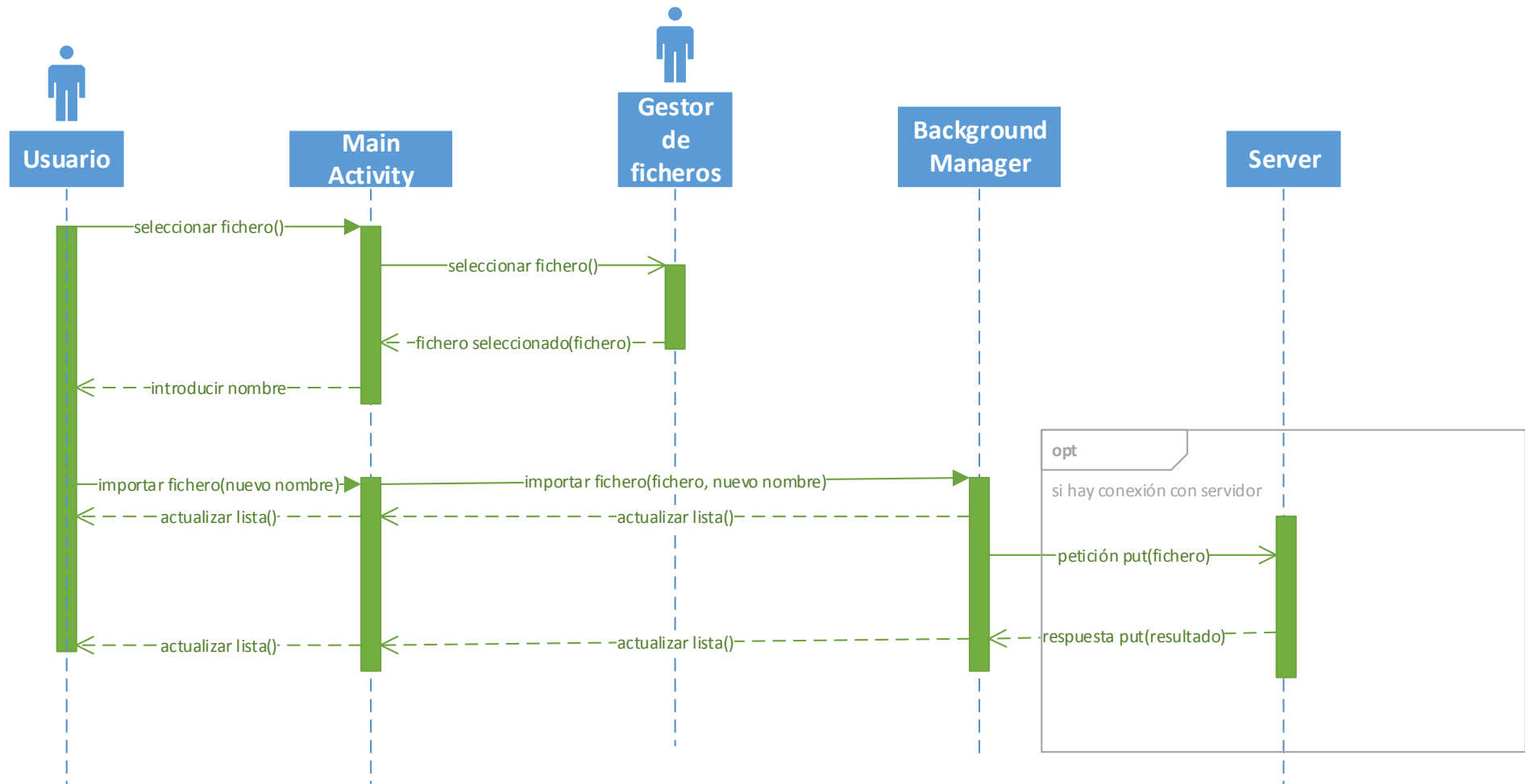


Ilustración 17. Diagrama de secuencia. Importar fichero

4.3.4. Abrir fichero

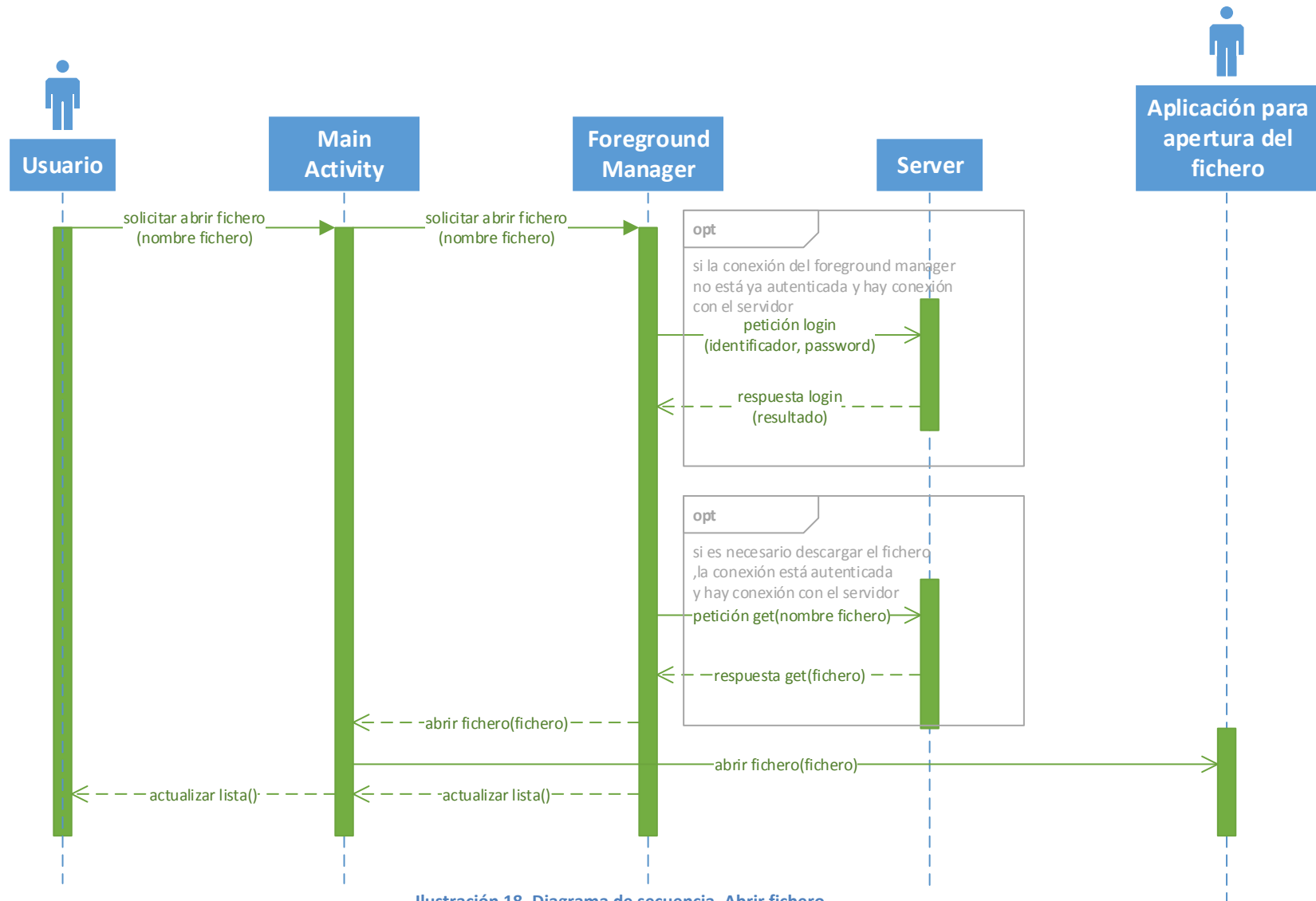


Ilustración 18. Diagrama de secuencia. Abrir fichero

4.3.5. Editar fichero

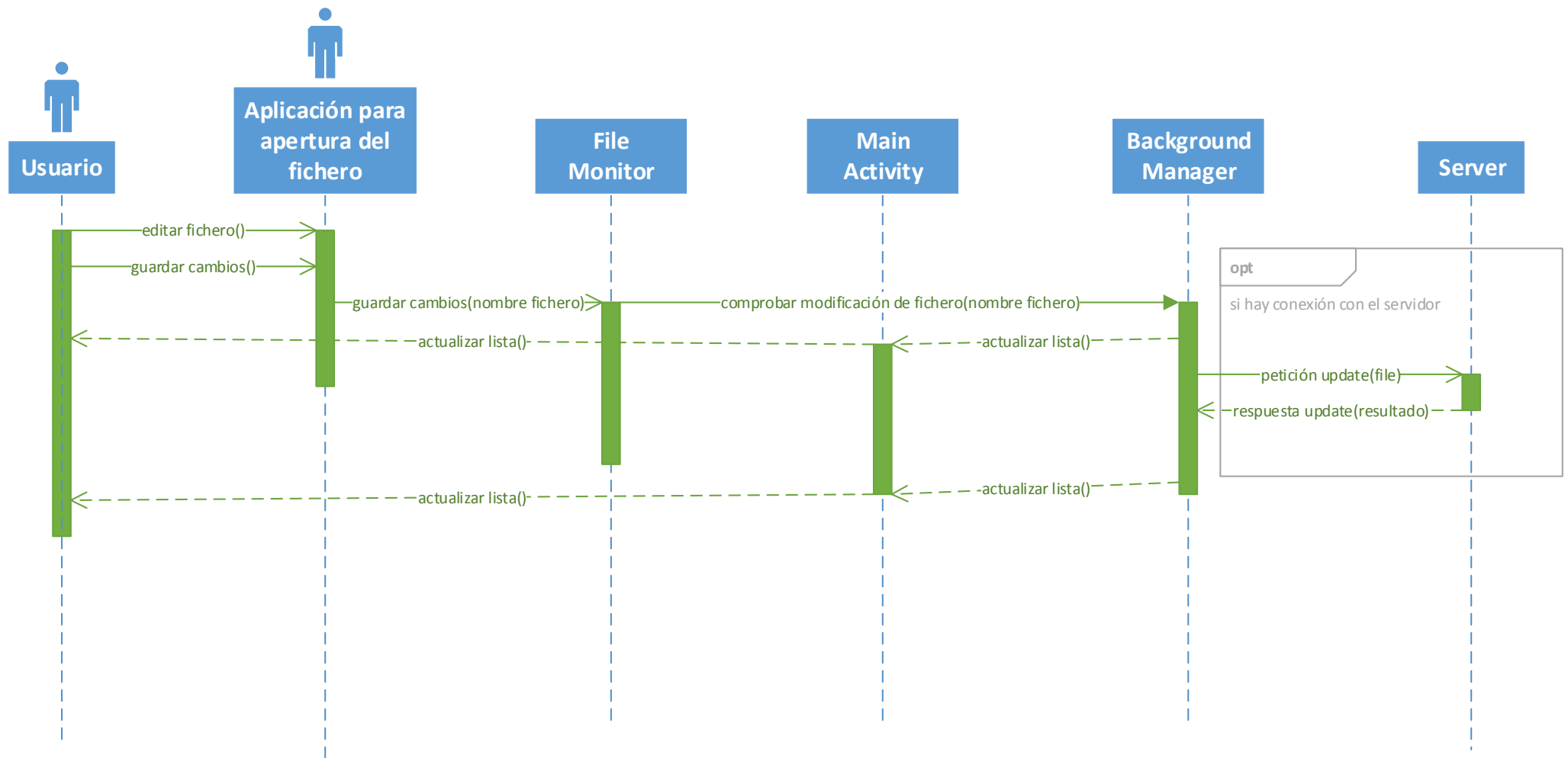


Ilustración 19. Diagrama de secuencia. Editar fichero

4.3.6. Borrar fichero

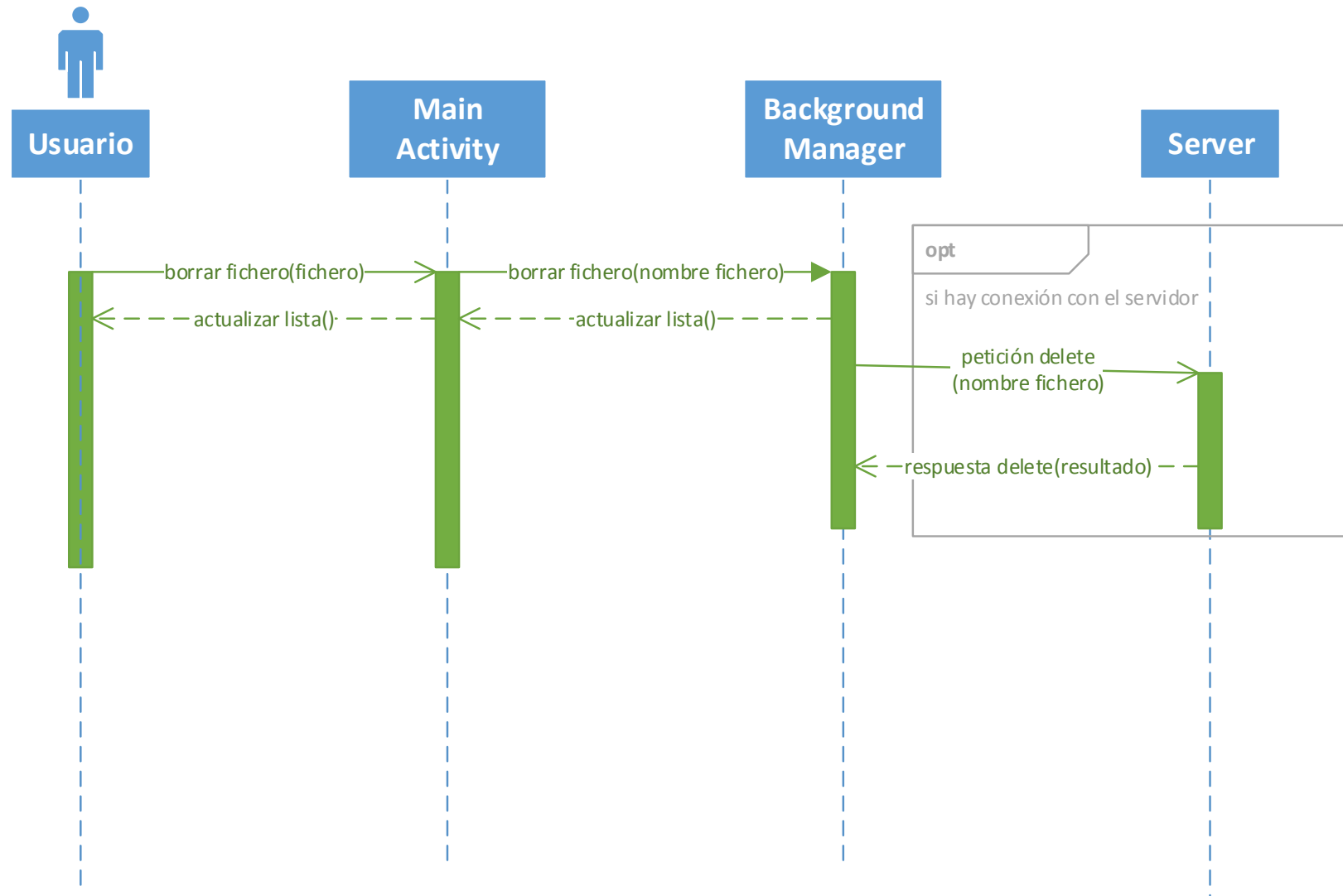


Ilustración 20. Diagrama de secuencia. Borrar fichero

4.3.7. Modificar etiquetas

Este diagrama especifica las interacciones correspondientes a los casos de uso añadir etiqueta y quitar etiqueta. Se ha decidido juntar los dos casos de uso en un solo diagrama de secuencia debido a que ambos producen la misma secuencia de interacciones, ya que la modificación de las etiquetas no se lleva a cabo hasta que se sale de la pantalla donde se muestran las etiquetas. Por tanto todas las modificaciones realizadas en las etiquetas de un fichero se efectúan en ese momento, sean adiciones o eliminaciones y sea una sola o varias modificaciones.

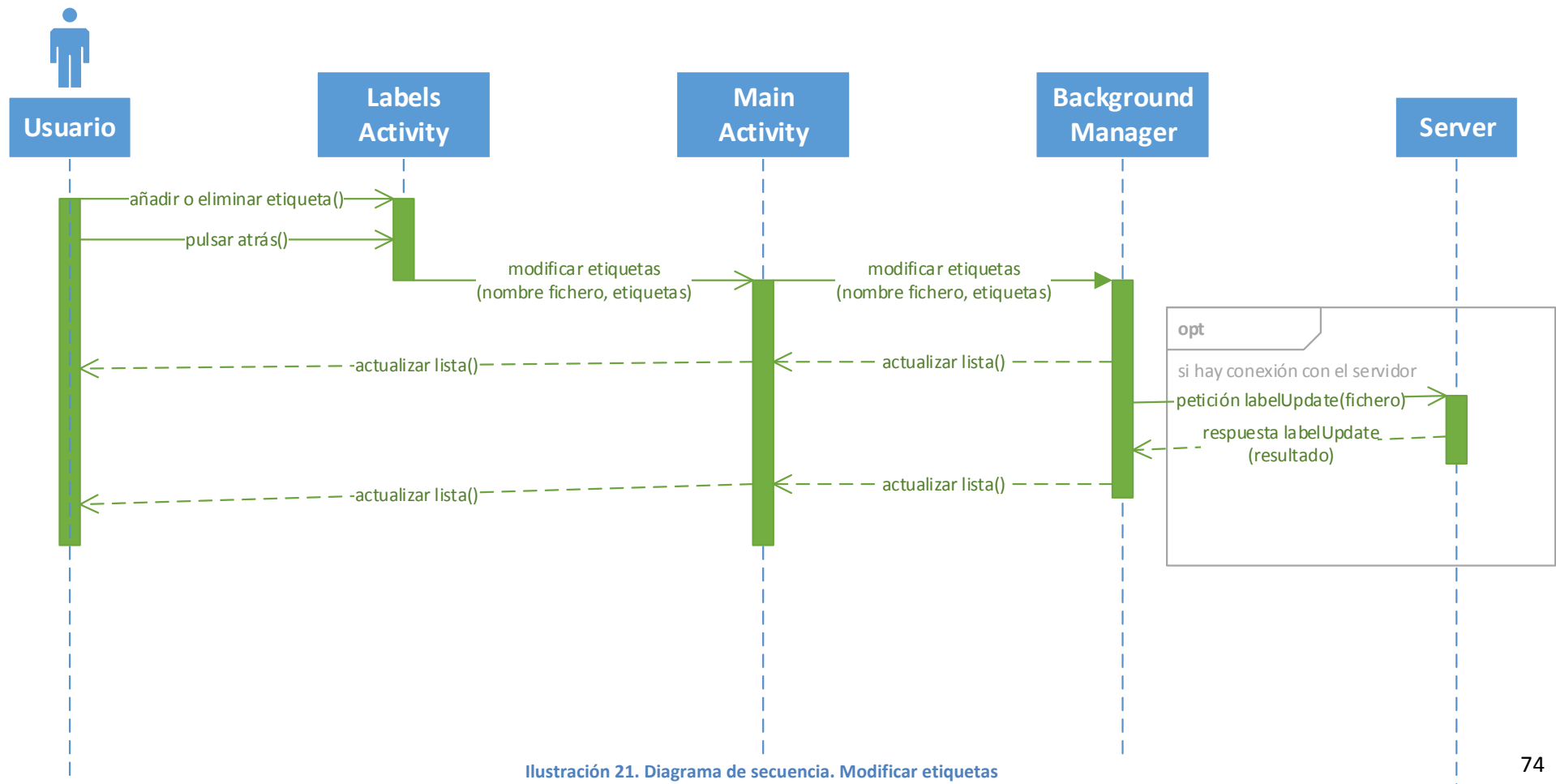


Ilustración 21. Diagrama de secuencia. Modificar etiquetas

4.3.8. Filtrar ficheros por etiqueta

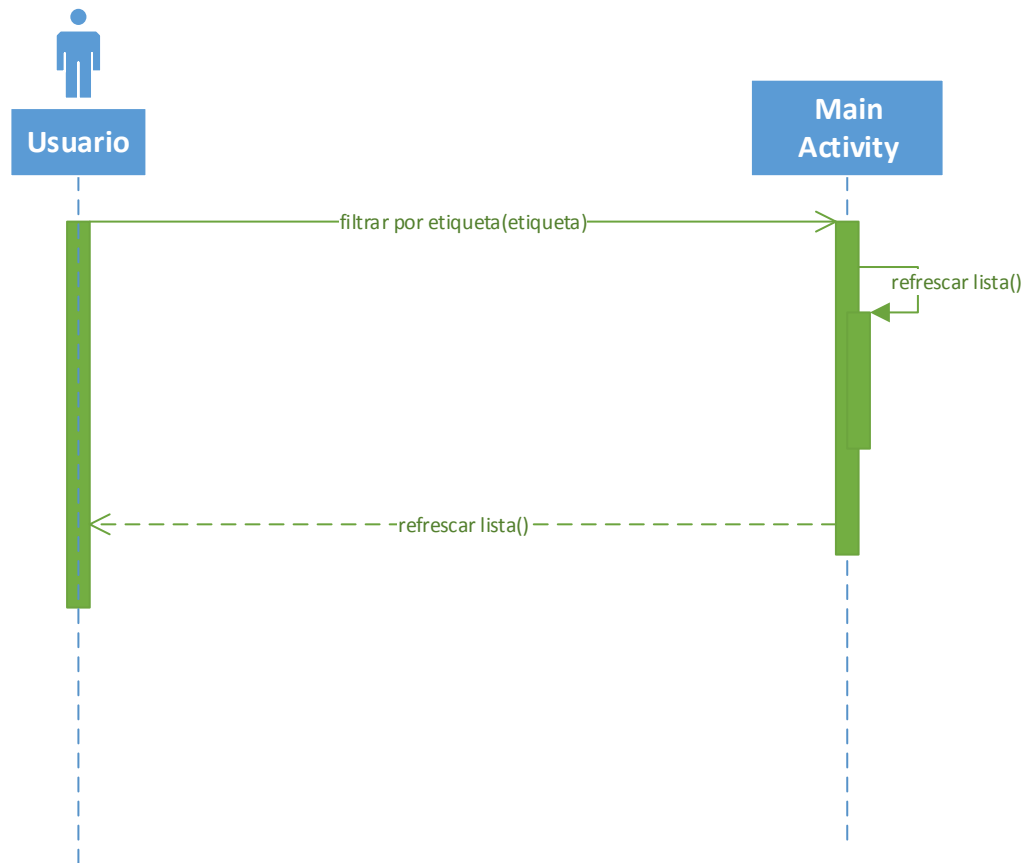


Ilustración 22. Diagrama de secuencia. Filtrar ficheros por etiqueta

5. IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se explican los aspectos relativos a la implementación del sistema. Para ello se describe el protocolo de comunicación entre cliente y servidor, el formato de los ficheros utilizados en el sistema, la codificación de los datos utilizada, los sockets e hilos empleados y otras consideraciones de la implementación.

En primer lugar habría que recordar las diferentes categorías de datos existentes, que ya se especificaron en la sección de arquitectura del sistema. Por una parte tenemos los ficheros de datos propiamente dichos y por otra parte los metadatos. Algunos metadatos aparecen tanto en los ficheros de metadatos como en el protocolo de comunicación pero además existen ciertos campos que son específicos del protocolo, que se explicarán más adelante.

Cabe destacar que tanto el formato de los datos y los metadatos (la organización de los distintos campos) como su codificación (la representación binaria de los mismos) son iguales en el protocolo de comunicación y en los ficheros utilizados. Con esto se consigue una convergencia de ambas funciones que posibilita la reutilización del código utilizado para ambas tareas.

5.1. Protocolo de comunicación

En esta sección se describe el protocolo de comunicación entre la aplicación cliente y el servidor. El protocolo es el que describe el formato de los mensajes intercambiados entre ambos, es decir, la organización de los diferentes campos en los mensajes.

Para el desarrollo de este protocolo no se ha recurrido a ningún protocolo existente, por tanto se trata de un protocolo totalmente nuevo y específico para este sistema. El protocolo funciona directamente sobre TCP, es decir, funciona directamente sobre un socket de la capa de transporte.

Se analizaron algunos protocolos existentes como SOAP [18], tecnologías como *Web Services* [19] o técnicas REST [20], pero se descartaron debido a que introducen una sobrecarga adicional tanto en el procesamiento como en la cantidad de datos transmitidos. En el caso de las técnicas REST, esto se produce debido al uso de cabeceras HTTP [21] en los mensajes. En el caso de SOAP o cualquier tecnología basada en *Web Services*, además de las cabeceras HTTP o de otros protocolos intermedios, hay que tener en cuenta también la sobrecarga que supone la

encapsulación de los datos en XML, la cual introduce etiquetas y al tratarse de un protocolo pensado para el intercambio de datos textuales obliga a la transformación de los datos binarios (en este caso el contenido de los ficheros de datos de usuario) para adecuarse al juego de caracteres válidos del protocolo (por ejemplo usando una codificación Base64). Todo ello hace que el rendimiento de estos protocolos sea muy bajo ya que se aumenta significativamente tanto la cantidad de información que se ha de transmitir como la complejidad de procesamiento de los mensajes, incrementándose también el consumo de batería del dispositivo.

Por todo lo expuesto anteriormente se ha desarrollado un protocolo nuevo. Si bien podría decirse que está ligeramente inspirado en el protocolo HTTP y en las técnicas REST, no cumple todas sus características, por lo que no puede ser considerado específicamente un sistema que sigue los principios REST.

Es muy importante tener en cuenta que todas las fechas que aparecen en el protocolo son siempre fechas de servidor, nunca fechas de cliente. Esto se debe a que los relojes de los diferentes clientes pueden no estar sincronizados entre ellos y con respecto al reloj del servidor. Usando siempre fechas de servidor podemos asegurar que todas estas fechas son coherentes.

A continuación se especifica el protocolo mediante unos diagramas que describen la organización de los diferentes campos de cada mensaje. Cada campo aparece especificado por una caja con el nombre del mismo. Cada campo está separado del siguiente mediante un carácter separador de campos, aunque no se especifica en estos diagramas. Los campos son enviados en el orden en que aparecen especificados de izquierda a derecha. Una línea nueva significa que se ha incluido un carácter separador de cabeceras, pero se trata del mismo mensaje. Cuando hay varias cajas situadas unas encima de otras unidas por una línea vertical significa que existe una bifurcación, es decir, existen diferentes valores para un campo que pueden condicionar la organización de los campos que vienen detrás.

5.1.1. Especificación general de los mensajes

A continuación se especifica el formato general de los mensajes del protocolo, tanto el de petición (mensaje del cliente al servidor) como de la respuesta (mensaje del servidor al cliente).

Petición General

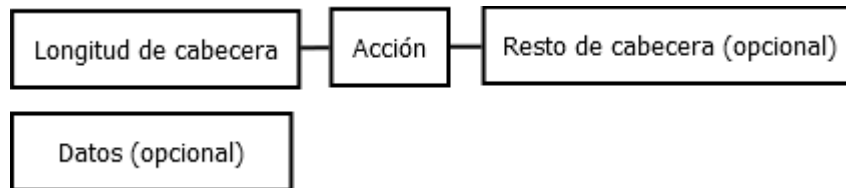


Ilustración 23. Protocolo. Petición General

Respuesta General

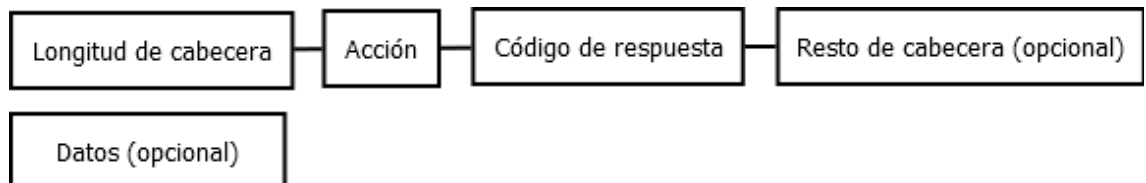


Ilustración 24. Protocolo. Respuesta General

La descripción de los campos de los diagramas son los siguientes:

- **Longitud de cabecera:** Indica la longitud de la cabecera del mensaje, esto es, la longitud del mensaje excluyendo los datos (contenido del fichero).
- **Acción:** Indica la acción que el cliente solicita que se realice, podría decirse que es una especie de identificador del tipo de petición. Es muy similar al llamado método o verbo del protocolo HTTP.

- **Código de respuesta:** indica el resultado de la petición, principalmente si la petición ha sido realizada correctamente o ha ocurrido algún error. Se envía en todas las respuestas. Sería muy similar al llamado código de respuesta o código de respuesta del protocolo HTTP.
- **Resto de cabecera:** es el resto de campos de la cabecera del mensaje. Es la parte que cambiará de un tipo de mensaje a otro. Hay mensajes que no incluyen esta parte del mensaje, es decir, que no incluyen más campos de los anteriormente descritos en la cabecera.
- **Datos:** es el contenido del fichero de datos, es decir, los datos binarios que se transmiten. Este campo sólo se incluye en determinados mensajes

Como se ha explicado anteriormente, todos los mensajes incluyen los campos especificados en los diagramas anteriores excepto los campos de resto de cabecera y datos que son opcionales

Las acciones del protocolo son las siguientes: Signup, Accountdelete, Login, List, Date, Get, Put, Update, Labelupdate y Delete.

Para realizar cualquier petición al servidor es necesario haber iniciado sesión previamente mediante un mensaje Login, excepto obviamente para realizar una petición de Signup, de Login o de Accountdelete.

En las siguientes secciones se describe en detalle el formato de los mensajes de cada tipo, que se corresponden con las distintas acciones, así como la utilidad de cada uno de ellos. Se describe tanto el mensaje de petición como el de respuesta. Para mayor claridad, en los diagramas no se incluye el campo de longitud de cabecera, pero como se ha especificado anteriormente éste se encuentra al principio de todo mensaje del protocolo.

5.1.2. Mensaje de Signup

El mensaje de Signup es el que solicita el registro de un nuevo usuario en el sistema. Obviamente para poder enviarlo no es necesario haber iniciado sesión previamente.

Petición

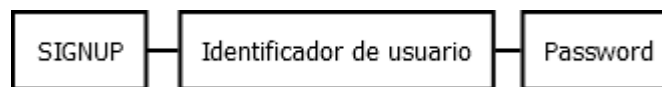


Ilustración 25. Protocolo. Petición Signup

Respuesta

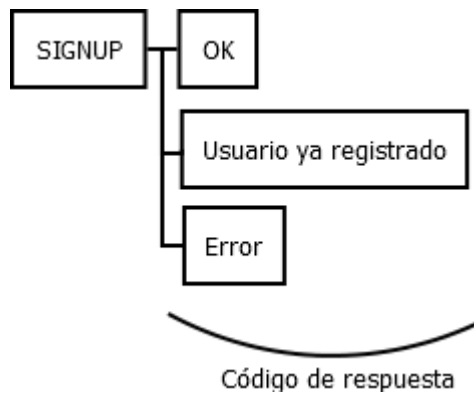


Ilustración 26. Protocolo. Respuesta Signup

5.1.3. Mensaje de Accountdelete

El mensaje de `Accountdelete` es el que solicita la eliminación de un usuario del sistema, es decir, se borra la cuenta del usuario incluyendo toda su información y sus ficheros. Para poder enviarlo no es necesario haber iniciado sesión previamente, ya que en el mismo mensaje se incluyen las credenciales de forma que se pueda asegurar que sólo el usuario pueda eliminar su propia cuenta.

Petición

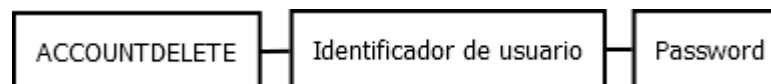


Ilustración 27. Protocolo. Petición Accountdelete

Respuesta

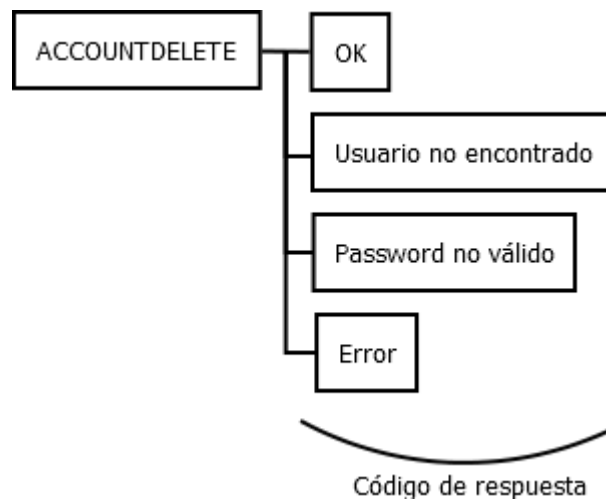


Ilustración 28. Protocolo. Respuesta Accountdelete

5.1.4. Mensaje de Login

El mensaje de `Login` es el que solicita el inicio de sesión de usuario en el sistema. Obviamente para poder enviarlo tampoco es necesario haber iniciado sesión previamente.

Petición

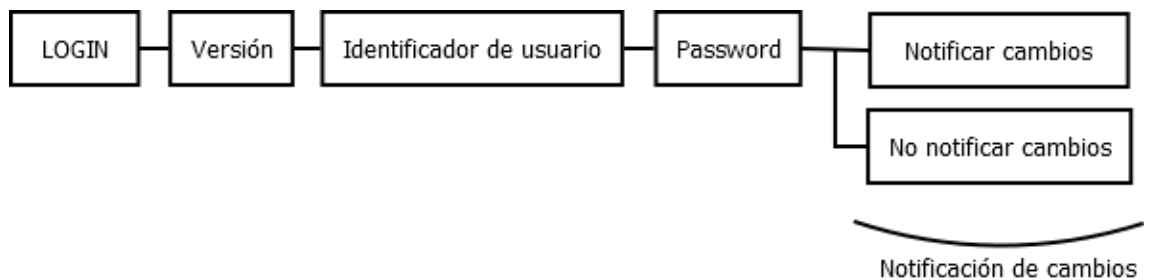


Ilustración 29. Protocolo. Petición Login

El campo versión se refiere a la versión de la aplicación cliente. Esto hace posible que si se modifica el cliente y se introducen nuevas versiones el servidor pueda adaptarse a las funcionalidades del cliente que se conecte, por ejemplo en caso de introducir funcionalidades nuevas o cambiar comportamientos de la aplicación.

El campo notificación de cambios especifica si se desea que el servidor envíe al cliente los cambios que se produzcan (adiciones, borrados y modificaciones de ficheros así como modificaciones de las etiquetas) en la cuenta del usuario a través de esta

conexión. Estas notificaciones se producirán mediante el envío de un mensaje de tipo Date desde el servidor al cliente.

Respuesta

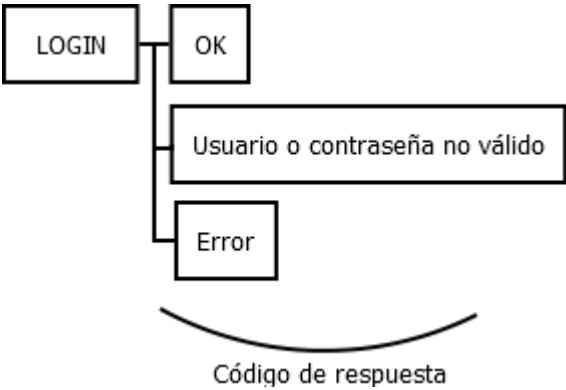


Ilustración 30. Protocolo. Respuesta Login

5.1.5. Mensaje de List

En el mensaje de `List` el cliente solicita la lista de ficheros que posee el usuario incluyendo todos sus metadatos. Con este mensaje el cliente recibe las actualizaciones y modificaciones de los metadatos de los ficheros para poder actualizar sus metadatos con la información más reciente del servidor, es decir, los metadatos del cliente se sincronizarán con los del servidor. Hay que tener en cuenta que no se transmite en ningún caso el contenido de ningún fichero, se trata únicamente de metadatos.

Petición

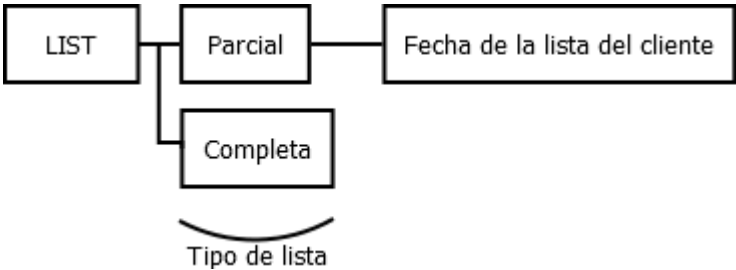


Ilustración 31. Protocolo. Petición List

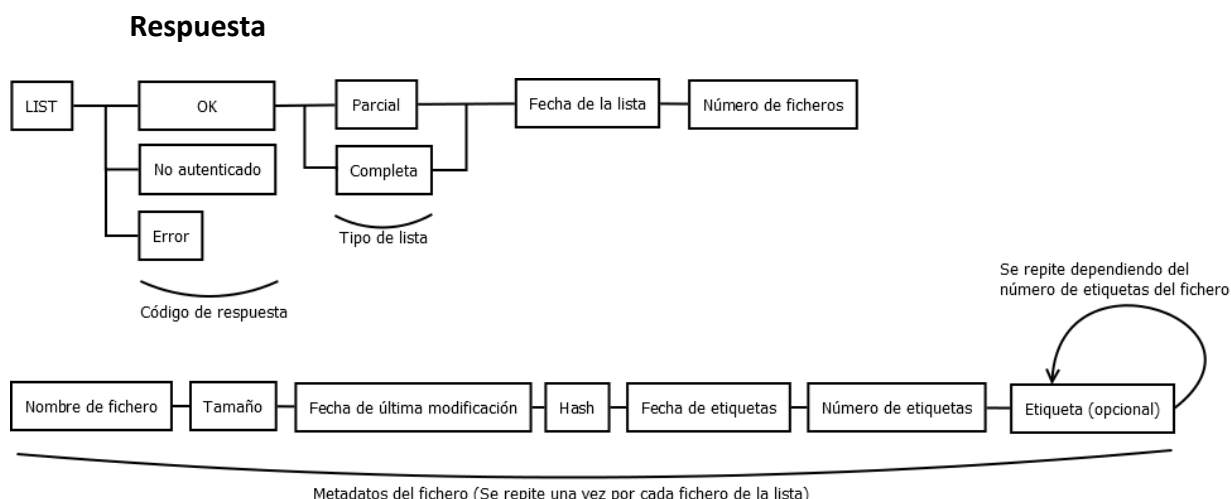


Ilustración 32. Protocolo. Respuesta List

Descripción de campos

A continuación se incluye una descripción de los campos del mensaje que pueden resultar confusos en el diagrama:

- Tipo de lista:** este campo en la petición indica si el cliente solicita una lista completa con todos los ficheros del usuario o al contrario una lista parcial que sólo incluya los ficheros que han sido añadidos o modificados (contenido o etiquetas) a partir de una determinada fecha que se envía en el campo “fecha de la lista del cliente”. En la respuesta, indica si el servidor envía una lista parcial o completa. Es necesaria esta especificación en la respuesta ya que aunque el cliente haya solicitado una lista parcial el servidor enviará una lista completa si se ha borrado algún fichero desde la fecha enviada en la petición. Esto se debe a que cuando se borra algún fichero se borran todos sus metadatos, por lo que la única forma de hacer al cliente saber que se ha borrado un fichero es enviar la lista completa. Los ficheros que se encuentran en el cliente pero que no aparezcan en la lista completa han sido borrados.
- Fecha de la lista del cliente:** este campo sólo se envía si el cliente solicita una lista parcial. Tomará el valor de la fecha de la última lista que recibió el cliente, por tanto en la respuesta sólo se van a recibir las actualizaciones que el cliente no conoce, en lugar de recibir los metadatos de todos los ficheros aunque no hayan cambiado desde entonces. Con ello se consigue un importante ahorro de la cantidad de datos transmitidos y que deberán ser procesados, sobre todo si el usuario posee gran cantidad de ficheros. Resumiendo, en este campo

indica la fecha a partir de la cual se desea recibir actualizaciones en la respuesta.

- **Fecha de la lista:** esta es la fecha de última modificación de la lista de ficheros en el servidor, sea por la creación de un nuevo fichero, una modificación de un fichero existente, por su borrado o por la modificación de sus etiquetas. Siempre será igual o posterior a la fecha más reciente de última modificación o de etiquetas de toda la lista de ficheros del servidor.
- **Fecha de última modificación:** es la fecha de última modificación del fichero en el servidor, o más bien, es la fecha de la última sincronización de un cambio del fichero en el servidor. Es muy importante tener en cuenta que se trata de fechas siempre de servidor, nunca fechas del cliente. Esto se debe a que los relojes de los diferentes clientes pueden no estar sincronizados entre ellos y con respecto al reloj del servidor. Usando siempre fechas de servidor podemos asegurar que todas estas fechas son coherentes.
- **Código de respuesta, no autenticado:** este código de respuesta indica que el cliente no se ha autenticado (no ha enviado un mensaje de Login), por lo que no se responde a la petición. El cliente debe autenticarse ante el servidor antes de realizar cualquier petición que no sea Login o Signup.
- **Número de ficheros:** es el número de ficheros que se envían en esta respuesta.
- **Hash:** es el hash del fichero.
- **Fecha de etiquetas:** es la fecha de última modificación de las etiquetas del fichero en el servidor.
- **Número de etiquetas:** es el número de etiquetas que posee el fichero.
- **Etiqueta:** es una etiqueta del fichero. Habrá un campo etiqueta por cada etiqueta que tenga el fichero.

Los campos incluidos dentro de la denominación metadatos del fichero en el diagrama representan los metadatos de cada uno de los ficheros que se envían en el mensaje. Toda esta secuencia de campos se repetirá tantas veces como indique el campo denominado número de ficheros.

5.1.6. Mensaje de Date

El mensaje de Date notifica al cliente que se ha producido algún cambio en la lista de ficheros del usuario, o lo que es lo mismo, se le notifica cuando se ha producido adiciones, borrados y modificaciones de ficheros así como modificaciones de las etiquetas. Para ello el mensaje Date envía la fecha de la lista de ficheros del cliente presente en el servidor (el campo fecha de la lista es equivalente al campo del mismo nombre en la respuesta List). El cliente puede comparar la fecha recibida con la que posee localmente y determinar si necesita actualizar su lista de ficheros mediante una petición List.

A diferencia del resto de los mensajes que siguen el modelo habitual cliente-servidor en el que el cliente envía una petición y el servidor envía una respuesta a dicha petición, el mensaje Date se envía desde el servidor al cliente sin necesidad de una petición previa del cliente, por lo que podría considerarse una respuesta sin petición. De esta forma se evita que el cliente necesite enviar peticiones periódicamente al servidor preguntando si se han producido cambios.

Mensaje Date



Ilustración 33. Protocolo. Mensaje Date

5.1.7. Mensaje de Get

En el mensaje de Get el cliente solicita el contenido de un determinado fichero, aunque en la respuesta también se incluyen sus metadatos. Este tipo de petición únicamente se enviará al servidor cuando un usuario solicite la apertura de un fichero y éste no se encuentre en el dispositivo cliente o se encuentre desactualizado. Esto es debido a que la descarga de ficheros es a petición del usuario.

Petición

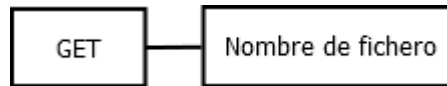


Ilustración 34. Protocolo. Petición Get

Respuesta

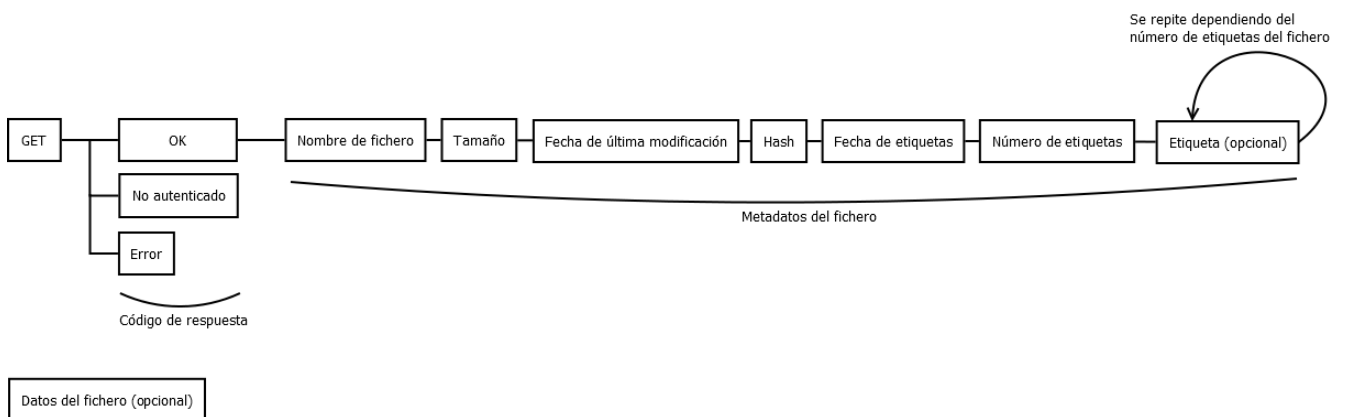


Ilustración 35. Protocolo. Respuesta Get

Descripción de campos

- **Datos del fichero:** Es el contenido del fichero de datos solicitado por el cliente. La longitud de este campo es igual al valor indicado por el campo de tamaño de los metadatos, por lo que si el tamaño es cero este campo no estará presente.

Los campos incluidos dentro de la denominación metadatos del fichero en el diagrama representan los metadatos del fichero y su formato es igual que en el mensaje de respuesta `List`.

5.1.8. Mensaje de Put

En el mensaje de `Put` el cliente solicita la creación de un nuevo fichero en el servidor. La petición incluye tanto los metadatos como el contenido del nuevo fichero, y en la respuesta se incluyen sus metadatos para que el cliente compruebe si éstos son correctos.

Petición

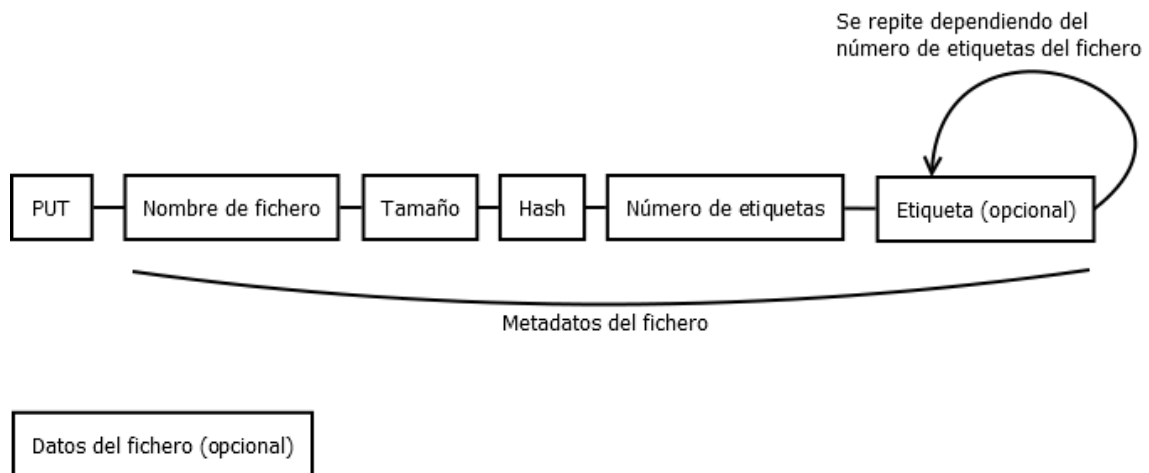


Ilustración 36. Protocolo. Petición Put

Respuesta

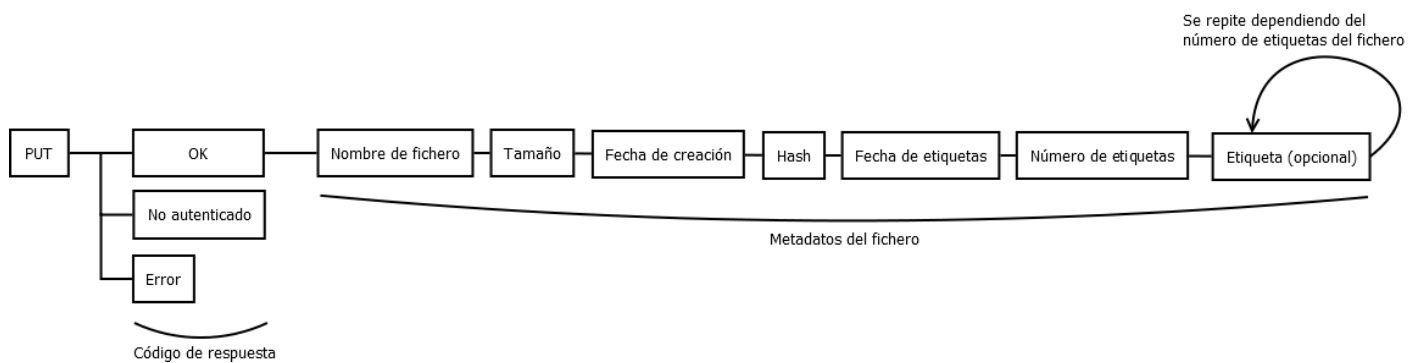


Ilustración 37. Protocolo. Respuesta Put

Descripción de campos

- **Fecha de creación:** es la fecha de creación del nuevo fichero en el servidor. Es análogo al campo llamado “fecha de última modificación” de otros mensajes como `Get` y `List`. De hecho cuando el cliente reciba esta fecha la almacenará como fecha de última modificación, ya que la semántica es la misma.
- **Fecha de etiquetas:** es el mismo campo de otros mensajes como `Get` y `List`, es decir, es la fecha de última modificación de las etiquetas del

fichero. En este caso toma la fecha de creación del fichero en el servidor.

- **Datos del fichero:** es el contenido del nuevo fichero de datos enviado por el cliente. La longitud de este campo es igual al valor indicado por el campo de tamaño de los metadatos, por lo que si el tamaño es cero este campo no estará presente.

5.1.9. Mensaje de Update

En el mensaje de `Update` el cliente solicita la sincronización en el servidor de un fichero que ha modificado, es decir, se suben los cambios realizados en el fichero al servidor. La petición incluye tanto los metadatos como el contenido del fichero modificado, y en la respuesta se incluyen sus metadatos para que el cliente compruebe si éstos son correctos.

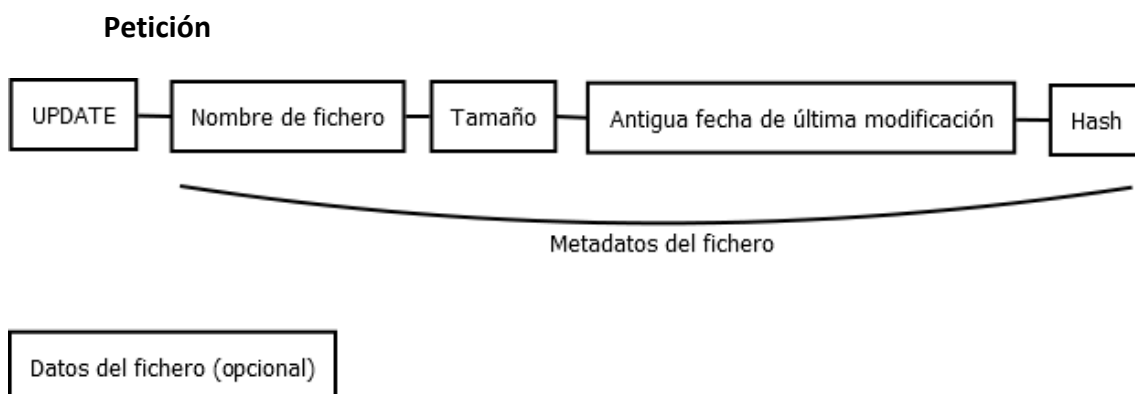


Ilustración 38. Protocolo. Petición Update

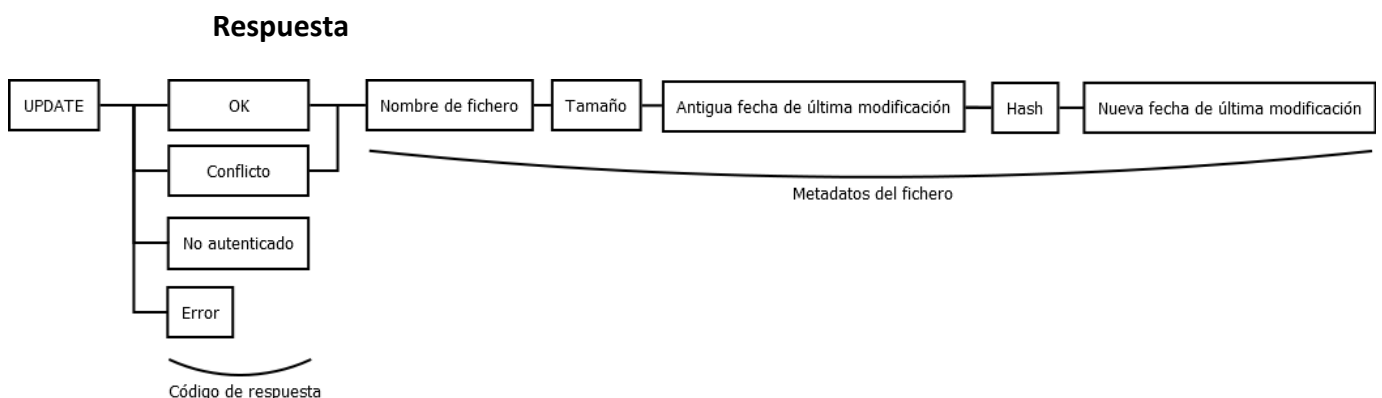


Ilustración 39. Protocolo. Respuesta Update

Descripción de campos

- **Antigua fecha de última modificación:** es la fecha de última modificación del fichero en el servidor, o más bien, es la fecha de la última sincronización de un cambio del fichero en el servidor. Es análogo al campo llamado “fecha de última modificación” de otros mensajes como `Get` y `List`. En este caso esta fecha es la que tiene el cliente y el servidor antes de que la modificación de esta petición `Update` se haya producido, por eso en este caso se ha denominado “antigua”. Esta fecha es la que tiene almacenada el cliente de la última vez que actualizó sus metadatos (por ejemplo al recibir una respuesta `List`). El servidor comprueba que el valor de este campo que recibe coincide con el valor de fecha de última modificación del fichero que él mismo tiene en sus metadatos. De esta forma se asegura de que la modificación del fichero se ha realizado sobre la versión actual del fichero en el servidor.
- **Nueva fecha de última modificación:** es la nueva fecha de última modificación que adquiere el fichero después de que se realiza la petición `Update`. Cuando el cliente la recibe la respuesta debe actualizar en sus metadatos el valor de la fecha de última modificación del fichero al valor que tiene este campo del mensaje.
- **Datos del fichero:** es el contenido del fichero de datos modificado enviado por el cliente. La longitud de este campo es igual al valor indicado por el campo de tamaño de los metadatos, por lo que si el tamaño es cero este campo no estará presente.

Como ya se ha dicho anteriormente es muy importante recordar que todas las fechas que aparecen en el protocolo son siempre fechas de servidor, nunca fechas de cliente. Esto se debe a que los relojes de los diferentes clientes pueden no estar sincronizados entre ellos y con respecto al reloj del servidor. Usando siempre fechas de servidor podemos asegurar que todas estas fechas son coherentes.

5.1.10. Mensaje de `Labelupdate`

En el mensaje de `Labelupdate` el cliente solicita la sincronización en el servidor de la modificación de las etiquetas de un fichero, es decir, se suben los cambios realizados en las etiquetas del fichero al servidor. El funcionamiento de este mensaje es bastante similar al mensaje `Update`, con la única diferencia de que

mientras en aquel se busca sincronizar modificaciones en el contenido del fichero, en el mensaje `Labelupdate` sólo se busca sincronizar modificaciones en las etiquetas del mismo.

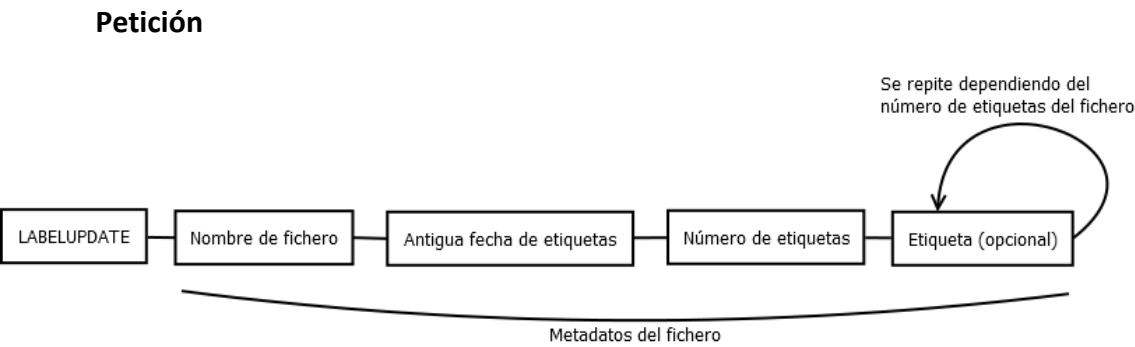


Ilustración 40. Protocolo. Petición `Labelupdate`

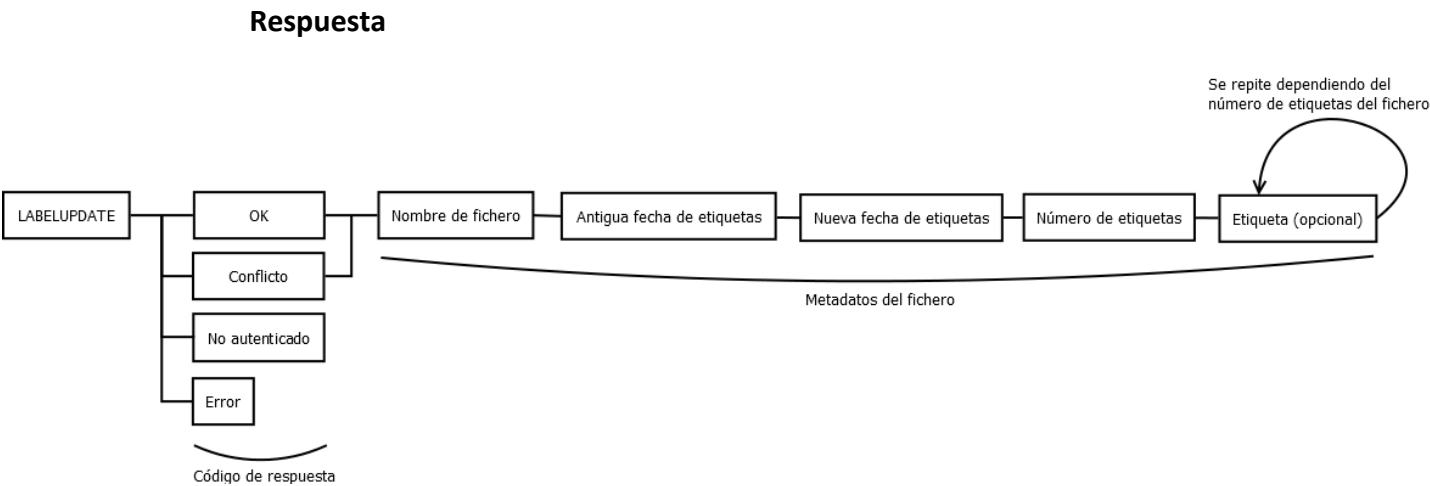


Ilustración 41. Protocolo. Respuesta `Labelupdate`

Descripción de campos

- **Antigua fecha de etiquetas:** es la fecha de última modificación de las etiquetas del fichero en el servidor, o más bien, es la fecha de la última sincronización de un cambio en las etiquetas del fichero en el servidor. Su funcionamiento es muy similar al campo “antigua fecha de última modificación” de los mensajes `Update`. Es análogo al campo llamado “fecha de etiquetas” de otros mensajes como `Get` y `List`. En este caso esta fecha es la que tiene el cliente y el servidor antes de que la modificación de esta petición `Labelupdate` se haya producido, por

eso en este caso se ha denominado “antigua”. Esta fecha es la que tiene almacenada el cliente de la última vez que actualizó sus metadatos (por ejemplo al recibir una respuesta `List`). El servidor comprueba que el valor de este campo que recibe coincide con el valor de fecha de etiquetas del fichero que él mismo tiene en sus metadatos. De esta forma se asegura de que la modificación de las etiquetas del fichero se ha realizado sobre la versión actual de las etiquetas del fichero en el servidor.

- **Nueva fecha de etiquetas:** es la nueva fecha de etiquetas que adquiere el fichero después de que se realiza la petición `Labelupdate`. Cuando el cliente la recibe la respuesta debe actualizar en sus metadatos el valor de la fecha de modificación del fichero al valor que tiene este campo del mensaje.

Al igual que ocurre con las fechas de última modificación del fichero, las fechas de etiquetas siempre son fechas de servidor para que éstas sean coherentes, como ya se ha explicado con anterioridad.

5.1.11. Mensaje de Delete

En el mensaje de `Delete` el cliente solicita el borrado de un fichero del servidor.

Petición

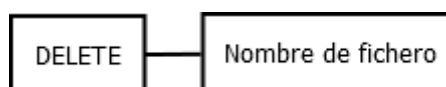


Ilustración 42. Protocolo. Petición Delete

Respuesta

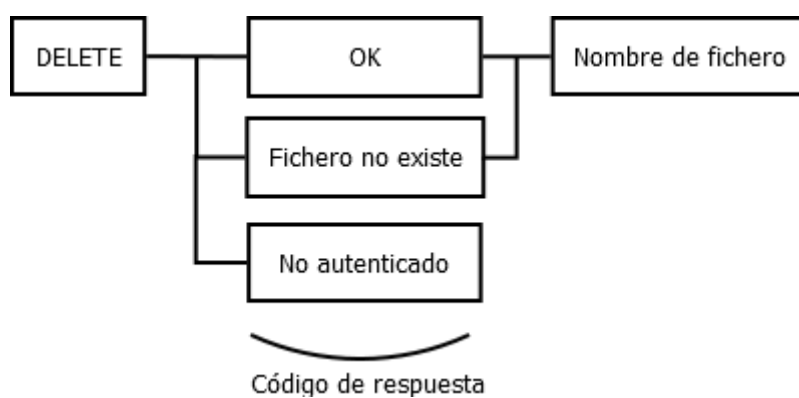


Ilustración 43. Protocolo. Respuesta Delete

5.2.Ficheros

En esta sección se explica cómo se almacena toda la información que maneja el sistema. Como en este sistema no se ha utilizado ninguna base de datos para este propósito toda la información se guarda en ficheros, tanto los datos como los metadatos y tanto en el cliente como en el servidor.

En primer lugar tenemos los ficheros de datos, que son los ficheros que los usuarios introducen en el sistema y su sincronización es el principal propósito del mismo. Estos ficheros se guardan en el almacenamiento persistente (disco en el caso del servidor y tarjeta de memoria del dispositivo cliente) tal cual los introduce el usuario, sin realizar ninguna transformación de los mismos. Cabe destacar que en el caso del dispositivo cliente, los ficheros solo se descargan bajo petición del usuario, con lo que se consigue un ahorro importante de espacio de almacenamiento en el dispositivo.

En el sistema existen también los ficheros de metadatos, que es donde se guarda toda la información necesaria para el funcionamiento del sistema. En el servidor existe un fichero de metadatos para cada usuario registrado en el sistema y en cada cliente existe únicamente un fichero de metadatos con la información del cliente que ha iniciado sesión en ese dispositivo. Es en estos ficheros donde se guarda toda la información relativa al usuario, como puede ser identificador y contraseña, así como la lista de ficheros que el usuario ha introducido en el sistema junto con metadatos sobre estos ficheros como son el tamaño, el hash, la fecha de última sincronización o las etiquetas. El formato en el que se guarda la información en estos ficheros de metadatos es similar al utilizado cuando se transmiten entre cliente y servidor mediante el protocolo anteriormente descrito. De esta forma se pueden reutilizar gran parte de los métodos de codificación y decodificación de mensajes para la lectura y escritura de estos ficheros. Además, al utilizarse el mismo formato y codificación se simplifica el procesamiento de todos los metadatos, ya que no es necesario realizar transformaciones adicionales en los mismos para almacenarlos o recuperarlos de los ficheros.

Además, cabe destacar que los ficheros de metadatos en el cliente y en el servidor son muy similares. Existen un conjunto de metadatos comunes a ambos que siguen un formato y codificación iguales, y determinados metadatos específicos en el cliente y el servidor que constituyen la única diferencia entre ambos. Estos metadatos específicos son los metadatos internos que se utilizan únicamente para el correcto funcionamiento de la aplicación cliente o servidor y no se transmiten a través del protocolo de comunicación.

A continuación se describen en detalle los ficheros de metadatos tanto del cliente como del servidor utilizando el mismo tipo de diagrama que el empleado para describir los mensajes del protocolo de comunicación. Los campos que son comunes al protocolo no se describirán aquí nuevamente, centrándose este apartado únicamente en la descripción de los metadatos internos, que son los que no aparecen en el protocolo.

5.2.1. Fichero de metadatos del servidor

A continuación se describe el formato del fichero de metadatos del servidor.

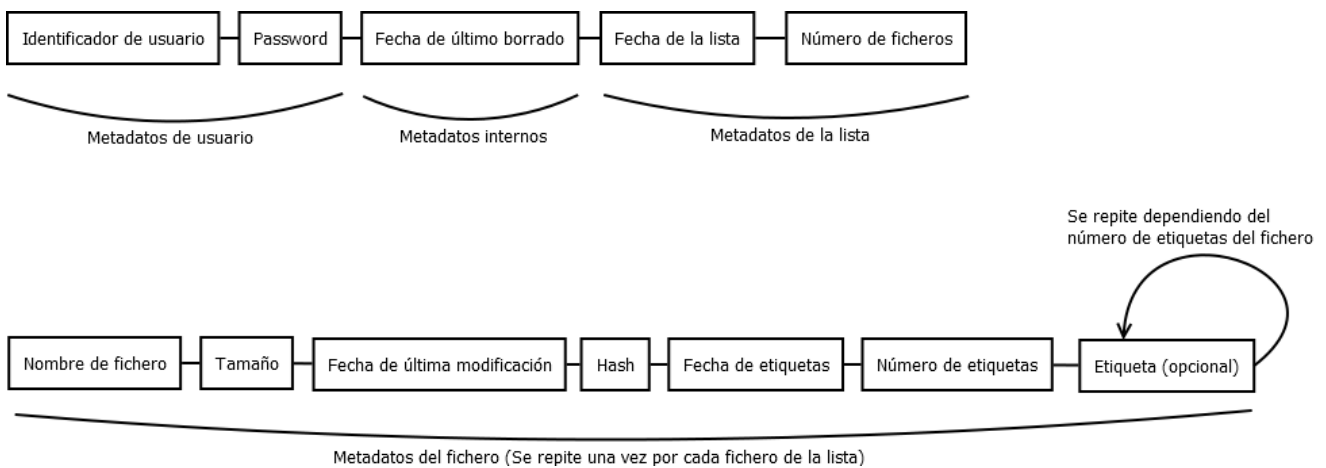


Ilustración 44. Fichero de metadatos del servidor

Descripción de campos de metadatos internos

- **Fecha de último borrado:** es la fecha de la última vez que se ha borrado un fichero en el servidor. Sirve al servidor para decidir si deberá enviar una lista completa cuando reciba una petición `List` del cliente. Como ya se ha explicado en la descripción del campo “tipo de lista” del mensaje de `List` del protocolo de comunicación, aunque el cliente haya solicitado una lista parcial, el servidor enviará una lista completa si se ha borrado algún fichero desde la fecha enviada en la petición, es decir, si la fecha del campo última fecha de borrado es posterior a la fecha enviada en la petición. Esto se debe a que cuando se borra algún fichero se borran todos sus metadatos, por lo que el servidor dispone de este campo para saber cuándo se borró un fichero por última vez. Gracias a este mecanismo el cliente sabe que los ficheros que no aparezcan en la lista completa han sido borrados.

Los campos marcados como metadatos de usuario son los metadatos relativos a los datos del usuario, los marcados como metadatos internos son los metadatos que necesita el servidor para funcionar correctamente pero que no se transmiten en el protocolo, los metadatos de la lista son los metadatos referidos a la lista de ficheros en general y los metadatos del fichero se refieren a cada uno de los ficheros de la lista (constituyen la lista de ficheros).

5.2.2. Fichero de metadatos del cliente

A continuación se describe el formato del fichero de metadatos del cliente.

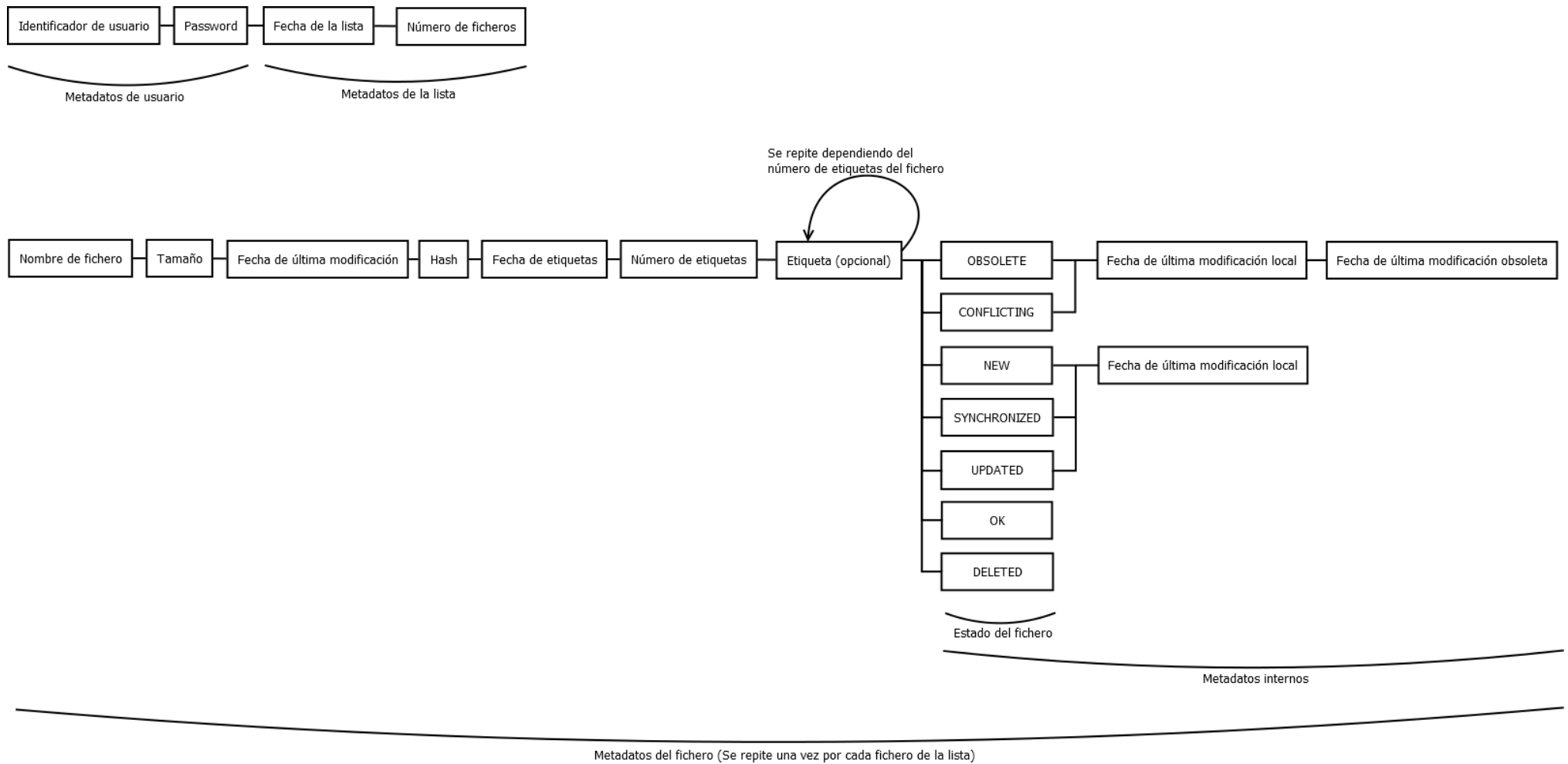


Ilustración 45. Fichero de metadatos del cliente

Descripción de campos de metadatos internos

- **Estado del fichero:** indica el estado del fichero en el cliente. Los posibles valores para este campo son los siguientes:
 - **OK:** el fichero no se encuentra físicamente en el dispositivo cliente (los ficheros sólo se descargan a petición del usuario).
 - **SYNCHRONIZED:** el fichero está presente físicamente en el dispositivo y su contenido se encuentra sincronizado, es decir, el cliente no conoce que haya otra nueva versión del mismo.
 - **OBSOLETE:** el fichero está presente físicamente en el dispositivo pero se trata de una versión obsoleta, es decir, el cliente conoce que existe una nueva versión del fichero pero el contenido del mismo en el cliente no está actualizado.
 - **NEW:** indica que el fichero es nuevo. El cliente lo usa para recordar que debe solicitar su creación en el servidor a través de una petición `Put`.
 - **UPDATED:** indica que el fichero ha sido modificado localmente en el cliente. Sirve al cliente para recordar que debe sincronizar los cambios en el servidor a través de una petición `Update`.
 - **CONFLICTING:** indica que existe un conflicto de versiones con el fichero. Al igual que `UPDATED` indica que el fichero ha sido modificado localmente en el cliente. El conflicto puede deberse a dos causas. La primera posible causa es que se ha modificado localmente una versión del fichero que el cliente ya conocía que estaba obsoleta (fichero con estado `OBSOLETE`). La otra posible causa es que se haya modificado la versión más reciente del fichero, pero otro cliente haya modificado esta misma versión y haya sincronizado las modificaciones en el servidor antes que este cliente. En cualquier caso, el conflicto de versiones se reduce al hecho de que se produzca una modificación de versiones no lineal o no incremental en el tiempo por diferentes clientes.
 - **DELETED:** indica que el fichero ha sido borrado. El cliente lo usa para recordar que debe solicitar el borrado del fichero en el servidor a través de una petición `Delete`.
- **Fecha de última modificación local:** indica cuál es la fecha de última modificación del fichero realizada localmente por el cliente. Se obtiene

del metadato de fecha de última modificación del fichero físico en el dispositivo, es decir, de la fecha proporcionada por el sistema de ficheros del dispositivo donde se ejecuta el cliente. Cuando el estado del fichero es `OK` o `DELETED` no se incluye este campo, ya que esto significa que el fichero no se encuentra físicamente en el dispositivo, y por tanto no existe esta fecha local. Es la única fecha que se usa en el sistema que es una fecha de cliente, no una fecha de servidor. En este caso el usar una fecha de cliente no presenta ningún inconveniente de falta de coherencia o sincronización de relojes puesto que esta fecha se usa únicamente internamente en el cliente y nunca se compara con otras fechas diferentes. Esta fecha sirve únicamente para ayudar a comprobar si el contenido del fichero ha sido modificado localmente (típicamente por alguna aplicación externa con la que se ha abierto previamente) sin tener que calcular siempre el hash del mismo (como se explicó en el requisito correspondiente).

- **Fecha de última modificación obsoleta:** es análogo al campo “fecha de última modificación” ya que guarda la fecha de última modificación en el servidor de la versión del fichero que hay en el dispositivo cliente. Se usa únicamente cuando el estado del fichero es `OBSOLETE` o `CONFLICTING`, ya que estos estados indican que el contenido del fichero no está actualizado a la última versión del servidor. Cuando se recibe una actualización de un fichero del servidor a través de una respuesta `List` indicando que el contenido del fichero ha cambiado, si el fichero se encuentra físicamente en el dispositivo cliente, entonces su contenido ya no estará actualizado a la nueva versión. Por ello se guarda el valor del campo “fecha de última modificación” que tenía el cliente en el campo “fecha de última modificación obsoleta” y se actualiza el campo “fecha de última modificación” con la nueva fecha enviada por el servidor. De esta forma se conserva la fecha de última modificación que realmente corresponde a la versión del contenido del fichero en el cliente.

Los campos marcados como metadatos internos son los metadatos que necesita el cliente para funcionar correctamente pero que no se transmiten en el protocolo. El resto de campos es igual que en el fichero de metadatos del servidor.

5.3.Codificación de datos

Los datos de la aplicación, es decir, el contenido de los ficheros de usuario no se codifica de ninguna forma específica en ningún momento, ni cuando se transmiten entre cliente y servidor (protocolo de comunicación) ni cuando se almacenan (ficheros de metadatos). Con esto se consigue minimizar tanto el procesamiento de los datos, como el espacio de almacenamiento requerido y la cantidad de información transmitida entre cliente y servidor.

En cuanto a la codificación de los metadatos, ésta es bastante sencilla para intentar minimizar el procesamiento requerido. Además busca minimizar también la cantidad de datos. Cabe destacar, que de igual forma que ocurre con la organización de los distintos campos de metadatos en el protocolo de comunicación cliente-servidor y la organización de los ficheros de metadatos utilizados, la codificación que se usa cuando los metadatos son serializados es también la misma cuando se transmiten entre cliente y servidor y cuando se almacenan en los ficheros de metadatos. Es decir, que tanto la organización de los distintos campos como la codificación de cada uno de ellos es igual en el protocolo y en los ficheros de metadatos. De esta forma se pueden reutilizar los algoritmos y el código para ambos propósitos.

La codificación de todos los campos de metadatos y de todos los campos de las cabeceras de los mensajes del protocolo de comunicación es de tipo textual, es decir, que todos los datos se codifican utilizando cadenas de texto (por ejemplo, los números se codifican con caracteres numéricos imprimibles). Esto se hace de esta forma ya que para separar los campos se utilizan determinados caracteres separadores y de esta forma estos datos puedan ser escaneados correctamente, por lo que los caracteres separadores no pueden aparecer dentro de ninguno de los campos (algo que podría ocurrir de manera fortuita si se introducen determinados tipos de datos sin codificar textualmente, como por ejemplo números codificados en binario).

En el protocolo de comunicación, debido al hecho de que los datos (contenido de los ficheros), no se codifican y se envían directamente en binario, es necesario un mecanismo para separar datos de metadatos, es decir, las cabeceras de los mensajes del contenido de los mismos (en caso de que lo tengan). Para ello se utiliza un campo de metadatos especial al principio de cada mensaje que indica la longitud de la cabecera del mensaje (metadatos), de forma que el sistema sabe hasta dónde puede escanear los campos usando los caracteres separadores. Si el tipo de mensaje incluye contenido (datos), el sistema entonces conoce la longitud de los datos ya que se incluye en uno de los campos de la cabecera (tamaño del fichero). En el caso de los ficheros de metadatos no es necesario este mecanismo de separación entre datos y metadatos ya que cada uno se guarda en un fichero diferente.

A continuación se describen más en detalle las codificaciones de los campos de metadatos utilizados en el sistema:

- **Cadenas de texto:** los campos representados por cadenas de texto no necesitan ningún otro tipo de codificación. Se codifican de esta manera los siguientes campos: nombres de fichero, etiquetas, identificadores de usuario y passwords. El único requisito es que no contengan ninguno de los caracteres separadores.
- **Fechas:** cada fecha se codifica como una cadena de texto que representa un número entero que es el número de milisegundos desde el 1 de Enero de 1970 a las 00:00:00 GMT. Esta codificación numérica ocupa menos caracteres que una codificación de fecha normal y además su procesamiento es más sencillo al no depender de la codificación de los campos de la fecha utilizados.
- **Números enteros:** se codifican como una cadena de texto que representa el número. Se codifican de esta manera los siguientes campos de metadatos: tamaño de fichero, número de etiquetas, número de ficheros y versión de la aplicación cliente.
- **Hashes:** cada hash se envía codificado en Base64 de forma que no se produzcan colisiones con los caracteres separadores. La codificación Base64 es computacionalmente costosa y además aumenta la longitud de la información codificada pero hay que tener en cuenta que esto no supone un gran problema en este sistema ya que la longitud de los hashes es muy limitada.

En cuanto al protocolo de comunicación cabe destacar que como se ha indicado anteriormente, la codificación de los campos de metadatos del protocolo es idéntica a la codificación en los ficheros de metadatos. A continuación se describen las codificaciones de los campos específicos del protocolo de comunicación, que se suman a los explicados anteriormente:

- **Longitud de cabecera:** es el único campo que no se codifica en forma de texto. No presenta problemas de colisiones con los caracteres separadores debido a que es de longitud fija y es lo primero que se decodifica del mensaje, antes de empezar a escanear los distintos campos. Se codifica como un número entero de 4 bytes directamente siguiendo la codificación binaria que posee el tipo int en Java.

- **Acciones del protocolo:** se codifican como una cadena de texto que representa un número entero. Cada acción se representa por un número diferente a modo de código de acción. Esta forma de codificación ocupa menos espacio que codificar cada acción con su nombre (LOGIN, LIST, UPDATE, etc.).
- **Códigos de respuesta:** se codifican de la misma forma que las acciones de protocolo. Cada código de respuesta se representa con un número entero diferente que lo identifica codificado en una cadena de texto.
- **Tipo de lista:** se codifica de la misma forma que las acciones de protocolo. Cada tipo de lista se representa con un número entero diferente que lo identifica codificado en una cadena de texto.
- **Estado del fichero:** se codifica de la misma forma que las acciones de protocolo. Cada estado se representa con un número entero diferente que lo identifica codificado en una cadena de texto.

5.4.Sockets e hilos del sistema

En esta sección se explicarán tanto los sockets empleados en el sistema para la conexión entre cliente servidor como los hilos empleados tanto en la aplicación servidor como en la aplicación cliente.

5.4.1. Sockets del sistema

Como ya se ha explicado en la especificación del protocolo de comunicación entre cliente y servidor, la conexión entre ambos se realiza directamente a través de un socket TCP, utilizando el protocolo creado específicamente para este sistema.

Cabe destacar que realmente se utilizan dos sockets TCP para conectar cada aplicación cliente con el servidor: el socket en primer plano y el socket en segundo plano.

El socket en primer plano se encargará de las transmisiones de información que afectan directamente a la interacción de usuario, es decir, a través de ésta conexión se envían los mensajes que afectan directamente a la interacción del usuario con la aplicación. La clase que se encarga de gestionar esta conexión en el cliente recibe el nombre de “ForegroundManager”, ya que podemos considerar que es la que se encarga de las transmisiones que se realizan en primer plano. Estas transmisiones son las relativas al control de las cuentas de usuario (como iniciar y cerrar sesión y crear y eliminar cuentas de usuario) y a la descarga de ficheros del servidor cuando el usuario solicita su apertura, ya que todas estas acciones deben obtener una respuesta lo más rápida posible del servidor para que el usuario espere lo mínimo posible.

El socket en segundo plano se encargará de las transmisiones de información que no afectan directamente a la interacción del usuario. Pueden considerarse como transmisiones en segundo plano (la clase que las gestiona en el cliente se denomina “BackgroundManager”) ya que no afectan directamente a la experiencia del usuario y además pueden realizarse sin la intervención del mismo. Estas transmisiones son las relativas al control de la sincronización entre cliente y servidor, es decir, son las que se encargan de sincronizar los cambios que se han producido en los ficheros (cambio en la lista de ficheros, creación, importación, borrado y modificación de ficheros y modificación de etiquetas).

5.4.2. Hilos de la aplicación servidor

En cuanto a los hilos utilizados en la aplicación servidor, existe un hilo principal que se encarga de estar a la escucha de nuevas conexiones. En el momento que se produce una nueva conexión, se crea un hilo específico que se encargará de atender las peticiones recibidas a través de esa conexión. El funcionamiento de la comunicación en la aplicación servidor es síncrono ya que el servidor recibe una petición del cliente, la procesa y acto seguido la responde, ocurriendo todo ello en el mismo hilo. Por tanto, como cada cliente establece dos sockets con el servidor, se crearán dos hilos en el servidor, uno para cada conexión.

Estos dos hilos creados en el servidor son prácticamente iguales, únicamente se limitan a recibir peticiones y enviar las respuestas correspondientes. La única diferencia que existe entre ellos es que a través de la conexión en segundo plano se enviará al cliente un mensaje para notificar que la lista de ficheros ha cambiado (mensaje `Date`, que es el único tipo de mensaje que se envía del servidor al cliente sin necesidad de una petición del cliente), mientras que este mensaje no se envía a través de la conexión en primer plano.

5.4.3. Hilos de la aplicación cliente

En la aplicación cliente la configuración de los hilos es bastante más compleja que en el servidor.

El cliente cuenta con dos componentes principales: una actividad principal y un servicio. Ambos son componentes básicos de una aplicación Android.

Además, cabe destacar que a diferencia del servidor, en el cliente la recepción de mensajes de ambos sockets se realiza de forma asíncrona, es decir, el hilo que manda una petición no queda bloqueado esperando la respuesta. Para ello los mensajes recibidos del servidor se procesan en un hilo diferente. Por tanto contamos con un hilo para el socket en primer plano y otro para el socket en segundo plano.

La estructura de los hilos de la aplicación cliente es la siguiente:

- Actividad principal de la aplicación
 - Hilo principal de interfaz de usuario
 - Hilo de recepción de socket en primer plano

- Servicio de la aplicación:
 - Hilo de recepción de socket en segundo plano
 - Hilo de envío de petición de sincronización
 - Hilo de monitor de ficheros

Actividad de la aplicación: Es el componente de la aplicación que proporciona la interacción con el usuario. Una actividad siempre cuenta con un hilo principal que se encarga del procesamiento de la interfaz de usuario como se explica en el siguiente punto.

Hilo principal de interfaz de usuario: La actividad principal de la aplicación cliente cuenta con un hilo principal que es el que se encarga de realizar el procesamiento relacionado con la interfaz de usuario. Para no bloquear la interfaz de usuario y mejorar la experiencia del mismo no se debe sobrecargar este hilo. Por ello la aplicación cliente cuenta con más hilos para realizar el resto de procesamiento, especialmente las operaciones más largas o las que requieren esperar a la respuesta del servidor u otros eventos.

Hilo de recepción de socket en primer plano: Como ya se ha explicado anteriormente la recepción de mensajes es asíncrona. Este hilo es el que se encarga de esperar la recepción de los mensajes provenientes del servidor a través del socket en primer plano y de procesarlos (por ejemplo cuando se recibe un fichero del servidor a través de un mensaje `Get` se encarga de descargarlo y solicitar su apertura al hilo principal de interfaz de usuario).

Servicio de la aplicación: Es el componente de la aplicación que se encarga de realizar todo el procesamiento que podemos considerar no relacionado con la interfaz de usuario, es decir, el procesamiento en segundo plano. Esto se hace así para que estas operaciones se continúen realizando aunque la interfaz no esté visible en ese momento. De no ser así, en el momento que la interfaz de usuario desapareciese de la pantalla se podría detener todo el procesamiento que se esté realizando. En caso de que el usuario realmente desee cerrar completamente la aplicación y no se realice ningún procesamiento más, existe una opción en el menú principal que permite detener este servicio y todos los hilos y procesos de la aplicación, incluidos los de segundo plano.

Hilo de recepción de socket en primer plano: Este hilo es el que se encarga de esperar la recepción de los mensajes provenientes del servidor a través del socket en segundo plano y de procesarlos (por ejemplo cuando se recibe la confirmación de una modificación de un fichero del servidor a través de un mensaje `Update` se encarga de

actualizar su estado y solicitar la actualización de la lista de ficheros de la interfaz de usuario al hilo principal). Cada vez que se recibe una confirmación de la sincronización de una actualización realizada por el cliente (Put, Update, Delete o Labelupdate), se comprueba si existe alguna actualización más que sincronizar y se envía la petición correspondiente. Esto se hace así para que la sincronización de las actualizaciones se realice de forma secuencial, de forma que hasta que no se ha recibido una respuesta a una actualización no se manda la siguiente.

Hilo de envío de petición de sincronización: Este hilo se encarga de comprobar si hay alguna petición de sincronización en curso realizada al servidor a través del socket en segundo plano cuando el usuario ha realizado algún cambio en algún fichero o etiqueta del sistema. Si ya hay una petición en curso, este hilo muere sin realizar ninguna acción más, en cambio si no hay ninguna sincronización en curso este hilo envía la petición al servidor. Este hilo es necesario para iniciar la sincronización cuando no exista ninguna petición de sincronización en curso pero preservando la su carácter secuencial explicado en el párrafo anterior.

Hilo de monitor de ficheros: Este hilo se encarga de monitorizar los eventos de escritura en los ficheros de forma que notifique a la aplicación en caso de que un fichero haya sido modificado, generalmente a través de una aplicación externa que permita la edición de este tipo de ficheros.

5.5. Otras consideraciones en la implementación

En esta sección se describen otras consideraciones que se han tenido en cuenta a la hora de implementar el sistema.

5.5.1. Modelo de metadatos del sistema

El modelo de metadatos del sistema, tanto en el cliente como en el servidor, se mantiene en memoria íntegramente mientras el sistema está funcionando, y sólo se vuelca en almacenamiento persistente (fichero de metadatos) cuando se va a eliminar de la memoria. Esto se hace así para evitar usar un sistema de base de datos y hacer el sistema más ligero.

5.5.2. Sistema de control de versiones y conflictos de fichero

El sistema de fechas implementado permite detectar posibles conflictos de modificación concurrente de ficheros. Asegura que cuando se modifique un fichero siempre se haga sobre la última versión del mismo disponible en el servidor. Con esto se evita que se pueda modificar una versión antigua de un fichero, por ejemplo en un cliente que en ese momento no tenga conexión al servidor y no pueda conseguir la última versión del fichero, y después se sincronice esta modificación del fichero en el servidor y por tanto se pierdan datos de la versión que había en el mismo. Cuando se produce un conflicto de versiones de un fichero, la aplicación cliente se lo indica al usuario y le da la opción de elegir con qué versión del fichero quiere quedarse, la versión que se encuentra en el servidor o la que se encuentra en el cliente en ese momento. En caso de elegir la versión del servidor, se descarta la versión del cliente y en caso de elegir la versión del cliente, se sincroniza ésta en el servidor, de forma que se sobrescribe la versión existente en el mismo.

Con todo esto se consigue asegurar que sólo se sincronizan las modificaciones de ficheros realizadas sobre la versión del fichero disponible en el servidor (que es la que consideramos la versión estable del fichero), a no ser que el usuario decida lo contrario. Para conseguir que este sistema funcione correctamente las fechas de última modificación del fichero deben ser coherentes. Esto se consigue usando únicamente fechas de modificación en el servidor, es decir, fechas de sincronización del servidor, ya que los relojes de los clientes pueden no estar sincronizados con el reloj del servidor.

6. PRUEBAS

En este capítulo se van a especificar las pruebas de aceptación realizadas al sistema desarrollado para comprobar que funciona correctamente y se cumplen todos los requisitos especificados. En primer lugar se describirán los dispositivos empleados para realizar las pruebas, después se especificarán las pruebas de aceptación realizadas y por último se mostrará una matriz de trazabilidad que vincula cada requisito funcional con la prueba o pruebas que lo verifican.

6.1. Dispositivos de prueba

Para realizar estas pruebas se han utilizado varios dispositivos con sistema operativo Android para ejecutar la aplicación cliente. Entre estos dispositivos encontramos 3 dispositivos reales y 2 dispositivos virtuales, es decir, dispositivos ejecutados mediante el simulador proporcionado por el ADT. Se considera importante probar el cliente en dispositivos físicos ya que de esta forma nos aseguramos que funcionará correctamente en dispositivos reales. Además, cabe destacar que uno de los dispositivos físicos utilizado tiene la versión de Android 2.2 (nivel de API 8), que es la versión mínima admitida por el sistema.

En las tablas siguientes se especifican los detalles de los diferentes dispositivos empleados en las pruebas.

Identificador	Dispositivo de prueba 1
Tipo de dispositivo	Dispositivo físico
Fabricante	Samsung
Modelo	Galaxy S II (GT-I9100P)
Versión de Android	4.1.2
Nivel de API	16
Procesador	Exynos Cortex A9 dual-core 1.2 GHz
Pantalla	480x800 píxeles de resolución y 4.3 pulgadas.
Memoria RAM	1 GB
Memoria dispositivo	1.97 GB
Memoria externa	11.50 GB (Almacén USB interno al dispositivo)

Tabla 44. Dispositivo de prueba 1

Identificador	Dispositivo de prueba 2
Tipo de dispositivo	Dispositivo físico
Fabricante	Sony
Modelo	Xperia Ray (ST17i)
Versión de Android	2.3.4
Nivel de API	10
Procesador	Qualcomm MSM8255 1GHz
Pantalla	480x854 píxeles de resolución y 3.3 pulgadas.
Memoria RAM	512 MB
Memoria dispositivo	1 GB
Memoria externa	4 GB (Tarjeta SD)

Tabla 45. Dispositivo de prueba 2

Identificador	Dispositivo de prueba 3
Tipo de dispositivo	Dispositivo físico
Fabricante	HTC
Modelo	Desire
Versión de Android	2.2
Nivel de API	8
Procesador	Qualcomm Snapdragon QSD8250 1 GHz
Pantalla	480x800 píxeles de resolución y 3.7 pulgadas.
Memoria RAM	576 MB
Memoria dispositivo	512 MB
Memoria externa	4 GB (Tarjeta SD)

Tabla 46. Dispositivo de prueba 3

Identificador	Dispositivo de prueba 4
Tipo de dispositivo	Simulador
Fabricante	N/A
Modelo	N/A
Versión de Android	2.3.3
Nivel de API	10
Procesador	N/A
Pantalla	480x800 píxeles de resolución y 4.0 pulgadas.
Memoria RAM	512 MB
Memoria dispositivo	200 MB
Memoria externa	500 MB (Tarjeta SD)

Tabla 47. Dispositivo de prueba 4

Identificador	Dispositivo de prueba 5
Tipo de dispositivo	Simulador
Fabricante	N/A
Modelo	Tablet
Versión de Android	4.4.2
Nivel de API	19
Procesador	N/A
Pantalla	1280x800 píxeles de resolución y 10.1 pulgadas.
Memoria RAM	512 MB
Memoria dispositivo	200 MB
Memoria externa	500 MB (Tarjeta SD)

Tabla 48. Dispositivo de prueba 5

En cuanto al dispositivo de prueba que se ha utilizado para ejecutar el servidor del sistema, no tiene demasiada relevancia debido a que no existe ninguna interfaz de usuario en el mismo. Al estar desarrollado en java, el único requisito para que funcione es que debe ejecutarse en una máquina que tenga una máquina virtual de java actualizada. Para realizar estas pruebas se han utilizado PCs con Windows 7, Windows XP y Ubuntu.

6.2. Pruebas de aceptación

A la hora de realizar las pruebas de aceptación hay que tener en cuenta que hay acciones que implican un cambio en el sistema que se debe sincronizar (crear, importar, modificar o borrar un fichero y añadir o eliminar etiquetas). Para realizar la verificación de que este tipo de pruebas se comprueban los resultados en un dispositivo diferente del utilizado para realizar la acción, pero en el cual el mismo usuario haya iniciado sesión. De esta forma nos aseguramos de que las acciones y cambios realizados no sólo se realizan en el dispositivo usado para realizar la acción, sino que se sincronizan al servidor y de allí al resto de dispositivos donde el usuario haya iniciado sesión.

Para la especificación de cada prueba de aceptación se utilizará el siguiente modelo de tabla:

Identificador	Título
Descripción	
Verificación	
Resultado	

Tabla 49. Plantilla para Pruebas de Aceptación

La descripción de cada uno de los campos de la tabla anterior es la siguiente:

- **Identificador:** Permite identificar unívocamente cada prueba de aceptación. Tiene la siguiente estructura: PA-XX. Consta de un prefijo PA que indica que se trata de una prueba de aceptación seguido de un número incremental único para prueba (XX del esquema anterior).
- **Título:** Es el nombre descriptivo asignado a la prueba.
- **Verificación:** Son las comprobaciones que se han realizado para verificar si la prueba se ha realizado correctamente.
- **Resultado:** Describe si la prueba ha sido exitosa o no.

Una vez descritos cada uno de los campos de la plantilla, se procede a exponer las tablas que describen cada prueba de aceptación.

PA-01	Registro de usuario
Descripción	Se registra un nuevo usuario en el sistema.
Verificación	Se comprueba que se han creado los directorios de datos del usuario en el servidor así como su fichero de metadatos.
Resultado	Éxito.

Tabla 50. PA-01. Registro de usuario

PA-02	Eliminación de cuenta de usuario
Descripción	Se elimina una cuenta de usuario del sistema.
Verificación	Se comprueba que se han eliminado los directorios de datos del usuario en el servidor así como su fichero de metadatos.
Resultado	Éxito.

Tabla 51. PA-02. Eliminación de cuenta de usuario

PA-03	Inicio de sesión de usuario
Descripción	Un usuario que se ha registrado previamente inicia sesión en el dispositivo.
Verificación	Se comprueba que se muestra la pantalla principal de la aplicación. Además, si la aplicación cliente se cierra y se vuelve a iniciar aparecerá directamente la pantalla principal en lugar de la pantalla de bienvenida (la cual permite al usuario registrarse o iniciar sesión).
Resultado	Éxito.

Tabla 52. PA-03. Inicio de sesión de usuario

PA-04	Cierre de sesión de usuario
Descripción	Un usuario que ha iniciado sesión previamente cierra su sesión en el dispositivo.
Verificación	Se comprueba que se muestra la pantalla de bienvenida (la cual permite al usuario registrarse o iniciar sesión) de la aplicación. Además, si la aplicación cliente se cierra y se vuelve a iniciar aparecerá la pantalla de bienvenida en lugar de la pantalla principal.
Resultado	Éxito.

Tabla 53. PA-04. Cierre de sesión de usuario

PA-05	Creación de nuevo fichero
Descripción	Un usuario crea un nuevo fichero en el sistema.
Verificación	Se comprueba que se ha creado un fichero vacío en los directorios de datos del usuario tanto en el dispositivo cliente como en el del servidor. Además el nuevo fichero aparecerá en la pantalla principal de la aplicación cliente. Se comprueba que el nuevo fichero aparece en la pantalla principal en otro dispositivo diferente en el que el mismo usuario haya iniciado sesión.
Resultado	Éxito.

Tabla 54. PA-05. Creación de nuevo fichero

PA-06	Importación de fichero
Descripción	Se importa un fichero existente en el dispositivo.
Verificación	Se comprueba que se ha añadido el fichero importado en los directorios de datos del usuario tanto en el dispositivo cliente como en el del servidor, y se verifica que el contenido es el mismo. Además el fichero importado aparecerá en la pantalla principal de la aplicación cliente. Se comprueba que el nuevo fichero aparece en la pantalla principal en otro dispositivo diferente en el que el mismo usuario haya iniciado sesión y se abre para comprobar que contiene la misma información que el fichero original.
Resultado	Éxito.

Tabla 55. PA-06. Importación de fichero

PA-07	Apertura de fichero
Descripción	Se abre un fichero que se encuentre en el sistema.
Verificación	Se comprueba que el fichero se abre en una aplicación externa y que el contenido del mismo es el correcto.
Resultado	Éxito.

Tabla 56. PA-07. Apertura de fichero

PA-08	Edición de fichero
Descripción	Un usuario modifica un fichero abierto en una aplicación externa y lo guarda.
Verificación	Se comprueba que el fichero en el servidor ha cambiado y contiene las modificaciones realizadas por el usuario. También se abre el fichero en otro dispositivo diferente en el que el mismo usuario haya iniciado sesión y se comprueba que el fichero contiene las modificaciones realizadas.
Resultado	Éxito.

Tabla 57. PA-08. Edición de fichero

PA-09	Borrado de fichero
Descripción	Un usuario borra un fichero existente en el sistema.
Verificación	Se comprueba que el fichero ha desaparecido de la pantalla principal de la aplicación en un dispositivo diferente en el que el mismo usuario haya iniciado sesión y además que el fichero ha desaparecido físicamente del dispositivo.
Resultado	Éxito.

Tabla 58. PA-09. Borrado de fichero

PA-10	Visualización de la lista de ficheros
Descripción	Cuando el usuario inicia la aplicación una vez que ha iniciado sesión en el dispositivo se debe mostrar la lista de ficheros.
Verificación	Se comprueba que se muestra la lista de ficheros del usuario en la pantalla principal de la aplicación cliente.
Resultado	Éxito.

Tabla 59. PA-10. Visualización de la lista de ficheros

PA-11	Visualización de etiquetas
Descripción	Un usuario requiere ver todas las etiquetas que posee un determinado fichero existente en el sistema.
Verificación	Se comprueba que se muestra la pantalla con todas las etiquetas que posee el fichero.
Resultado	Éxito.

Tabla 60. PA-11. Visualización de etiquetas

PA-12	Adición de etiqueta
Descripción	Un usuario añade una etiqueta nueva a un fichero existente en el sistema.
Verificación	Se visualizan las etiquetas del fichero en un dispositivo diferente en el que el mismo usuario haya iniciado sesión y se comprueba que se muestra la etiqueta añadida.
Resultado	Éxito.

Tabla 61. PA-12. Adición de etiqueta

PA-13	Eliminación de etiqueta
Descripción	Un usuario elimina una etiqueta de un fichero existente en el sistema.
Verificación	Se visualizan las etiquetas del fichero en un dispositivo diferente en el que el mismo usuario haya iniciado sesión y se comprueba que la etiqueta eliminada ha desaparecido.
Resultado	Éxito.

Tabla 62. PA-13. Eliminación de etiqueta

PA-14	Filtrado de ficheros por etiqueta
Descripción	Un usuario puede filtrar los ficheros que se muestran en la lista de ficheros basándose en que éstos posean una determinada etiqueta.
Verificación	Se comprueba que la lista de ficheros muestra los ficheros que contienen la etiqueta seleccionada.
Resultado	Éxito.

Tabla 63. PA-14. Filtrado de ficheros por etiqueta

PA-15	Eliminación de etiqueta del filtro
Descripción	Un usuario puede eliminar una etiqueta del filtro.
Verificación	Se comprueba que la lista de ficheros mostrados aumenta para incluir los ficheros que no tienen la etiqueta eliminada pero que poseen el resto de etiquetas del filtro.
Resultado	Éxito.

Tabla 64. PA-15. Filtrado de ficheros por etiqueta

PA-16	Desactivación del filtrado por etiqueta
Descripción	Un usuario desactiva el filtrado por etiqueta.
Verificación	Se comprueba que se muestran todos los ficheros existentes en el sistema en la lista de ficheros.
Resultado	Éxito.

Tabla 65. PA-16. Desactivación del filtrado por etiqueta

PA-17	Edición de ajustes
Descripción	Un usuario edita los ajustes del sistema.
Verificación	Se comprueba que los ajustes del sistema han sido modificados con los nuevos valores introducidos.
Resultado	Éxito.

Tabla 66. PA-17. Edición de ajustes

6.3. Matriz de trazabilidad Requisitos Funcionales – Pruebas

A continuación se muestra una matriz de trazabilidad que vincula cada requisito funcional con la prueba o pruebas que lo verifican, de forma que así se puede comprobar que todos y cada uno de los requisitos quedan verificados y probados.

PA RF	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
01	X																
02		X															
03			X														
04				X													
05					X												
06						X											
07							X										
08								X									
09									X								
10										X							
11											X						
12												X					
13													X				
14														X			
15															X		
16																X	
17																	X

Tabla 67. Matriz de trazabilidad Requisitos Funcionales - Pruebas

7. PLANIFICACIÓN Y PRESUPUESTO

En este capítulo se va a especificar en primer lugar la planificación del proyecto y después el presupuesto requerido para la realización del mismo.

7.1. Planificación

En esta sección se detalla la planificación seguida para la realización de este proyecto. La planificación se divide en tareas, especificándose para cada una de ellas la duración en días laborables, la fecha de comienzo y la fecha de fin. Las tareas se encuentran jerarquizadas; existen 4 tareas principales, que se corresponden a las 4 fases principales del desarrollo del proyecto que son: análisis, diseño, implementación y memoria. Cada una de estas 4 tareas principales se compone de varias subtareas.

Para la planificación se considera que todos los días son laborables a excepción de sábados y domingos. No se han considerado periodos vacacionales ni festivos. Las duraciones de las distintas tareas recogidas en la planificación consideran únicamente días laborables.

Cabe destacar que todas las tareas son realizadas por una única persona, el autor de este proyecto, por lo que no existe la posibilidad de realizar una tarea entre varias personas trabajando simultáneamente. Debido a ello la duración del proyecto se alarga considerablemente respecto de la duración conseguida por un equipo de desarrollo de varias personas.

A continuación se muestra una tabla con toda la información sobre la planificación del proyecto.

Nombre de tarea	Duración	Comienzo	Fin
Análisis	18 días	lun 13/05/13	mié 05/06/13
Toma de requisitos	8 días	lun 13/05/13	mié 22/05/13
Análisis del estado del arte	6 días	jue 23/05/13	jue 30/05/13
Elección de tecnologías	4 días	vie 31/05/13	mié 05/06/13
Diseño	24 días	jue 06/06/13	mar 09/07/13
Diseño del almacenamiento de metadatos	5 días	jue 06/06/13	mié 12/06/13
Diseño del protocolo de comunicación	6 días	jue 13/06/13	jue 20/06/13
Diseño del servidor	6 días	vie 21/06/13	vie 28/06/13
Diseño del cliente	7 días	lun 01/07/13	mar 09/07/13
Implementación	90 días	mié 10/07/13	mar 12/11/13
Implementación de la librería compartida	20 días	mié 10/07/13	mar 06/08/13
Implementación del servidor	25 días	mié 07/08/13	mar 10/09/13
Implementación del cliente Android	30 días	mié 11/09/13	mar 22/10/13
Pruebas y correcciones	15 días	mié 23/10/13	mar 12/11/13
Documentación del código	75 días	mié 10/07/13	mar 22/10/13
Memoria	22 días	mié 13/11/13	jue 12/12/13
Realización de la memoria	17 días	mié 13/11/13	jue 05/12/13
Revisión	5 días	vie 06/12/13	jue 12/12/13
TOTAL Proyecto	154 días	lun 13/05/13	jue 12/12/13

Tabla 68. Planificación de tareas del proyecto

A continuación se muestra la planificación en forma de un diagrama de Gantt, en el que se muestran las tareas de forma cronológica y donde se pueden observar con más claridad las dependencias existentes entre tareas. Como se puede observar cada tarea depende de su predecesora, es decir, cada tarea comienza cuando la tarea anterior finaliza. La única excepción a esta regla es la tarea número 15 “documentación del código”, la cual se realiza de forma simultánea a las actividades de implementación 11, 12 y 13. Esto se debe a que la documentación del código se realiza de forma simultánea a la implementación.

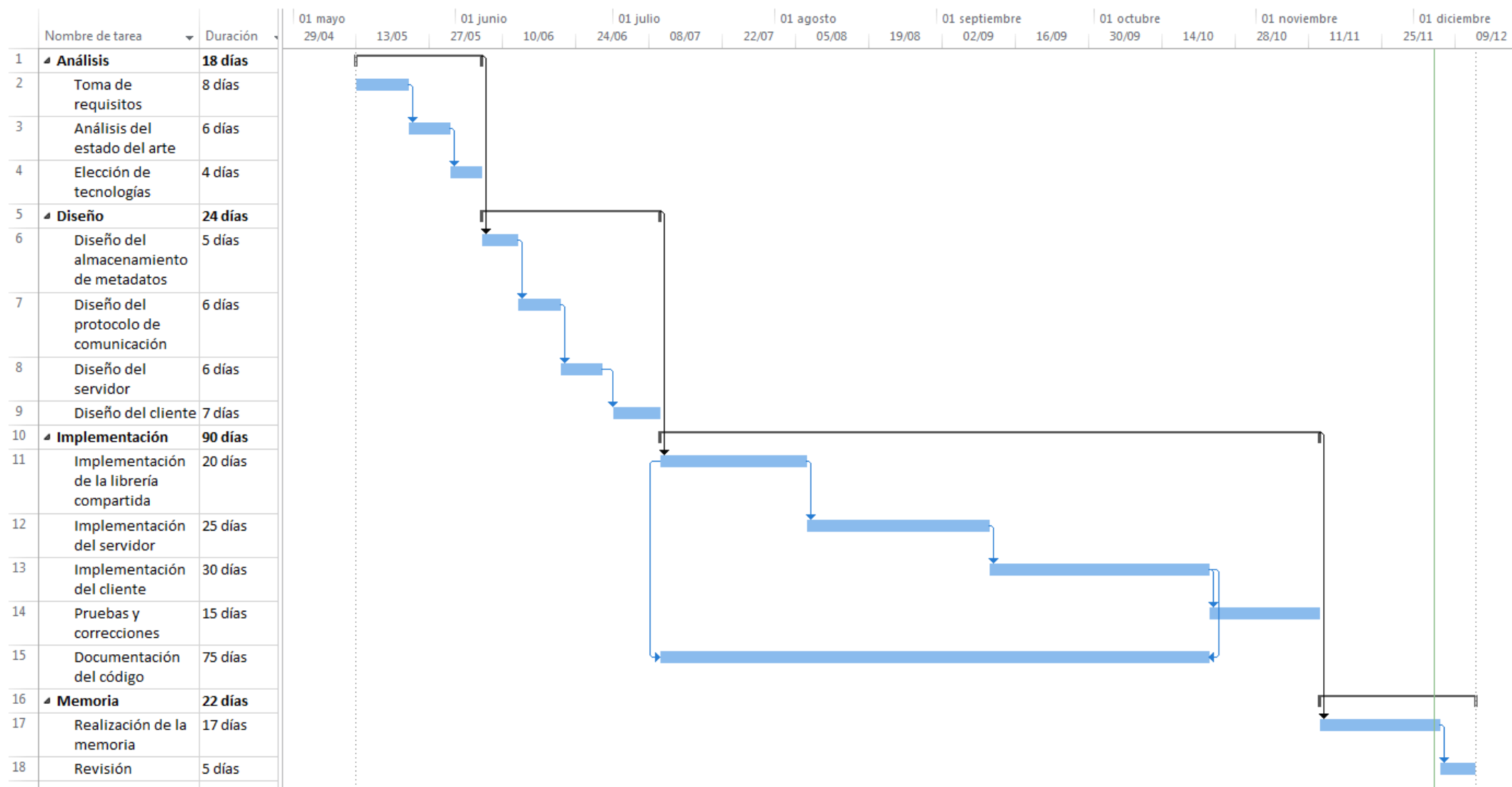


Ilustración 46. Diagrama de Gantt

7.2.Presupuesto

En esta sección se detalla el presupuesto del proyecto, es decir, la estimación de los costes de realización de este proyecto.

Todos los costes se mostrarán expresados en Euros.

La estimación de los costes se divide en costes de personal, costes de hardware y costes de software. No se incluye ningún cálculo de los costes de viajes, dietas o materiales fungibles ya que no se han necesitado en este proyecto. Por último se muestra un apartado con el cálculo del coste total del proyecto.

7.2.1. Costes de personal

En esta sección se calculan los costes asignados al personal del proyecto, es decir, el dinero dedicado a los profesionales que trabajan en el proyecto.

Para la estimación de los costes de personal se considera una jornada laboral de 8 horas. Esto unido a la duración de cada una de las tareas especificada en la planificación (que considera únicamente días laborables), nos permite calcular el número de horas reales de trabajo necesarias para completar cada una de las tareas. A partir del número de horas podemos calcular el coste de cada tarea.

Cabe destacar que aunque la categoría profesional necesaria para cada tarea puede cambiar, todas ellas son realizadas por el autor de este proyecto, es decir, que es la misma persona la que adopta todos los roles necesarios para el desarrollo del proyecto.

A continuación se incluye una tabla que muestra los costes de personal del proyecto desglosados por tareas y el total del proyecto. Las tareas que se muestran son las tareas principales especificadas en la planificación, a excepción de la tarea principal de implementación que aquí la hemos subdividido en dos subtareas debido a que cada una de ellas es realizada por un profesional diferente. Estas dos subtareas son:

- Implementación y documentación del código: Incluye las subtareas de implementación de la librería compartida, implementación del servidor, implementación del cliente Android y documentación del código.
- Pruebas y correcciones: Se corresponde con la subtaska del mismo nombre especificada en la planificación.

Costes de Personal					
Nombre de tarea	Duración (días)	Duración (horas)	Categoría	Coste por hora (€/hora)	Coste total (€)
Análisis	18	144	Analista	35	5040
Diseño	24	192	Diseñador	35	6720
Implementación y Documentación del código	75	600	Programador	25	15000
Pruebas y correcciones	15	120	Ingeniero de pruebas	30	3600
Memoria	22	176	Analista	35	6160
TOTAL Proyecto	154	1232			36520

Tabla 69. Costes de personal

7.2.2. Costes de hardware

En esta sección se especifican los costes asociados a los elementos hardware necesarios para el desarrollo del proyecto. Los conceptos incluidos en el cálculo incluyen el ordenador donde se ha realizado todo el desarrollo y donde se prueba el servidor, el dispositivo Android (Samsung Galaxy SII) donde se ha depurado y probado el cliente y los dispositivos Android extra donde se han realizado las pruebas del cliente.

Para calcular el coste imputable se ha utilizado la siguiente fórmula de amortización:

$$\text{Coste imputable} = \frac{\text{Dedicación}}{\text{Periodo de depreciación}} \times \text{Coste sin IVA} \times \% \text{ de uso}$$

Costes de Hardware					
Concepto	Coste sin IVA (€)	% Uso dedicado al proyecto	Dedicación (meses)	Periodo de depreciación (meses)	Coste imputable (€)
Dell Inspiron One 2350	908,27	80	7	60	84,77
Samsung Galaxy SII	453,41	50	7	60	26,44
Soxy Xperia Ray	329,75	100	1	60	5,49
HTC Desire	288,43	100	1	60	4,80
TOTAL Proyecto					121,52

Tabla 70. Costes de hardware

7.2.3. Costes de software

En esta sección se especifican los costes asociados a los elementos o licencias de software utilizados en este proyecto.

No se especifica aquí el coste del sistema operativo del ordenador de desarrollo debido a que viene preinstalado en el mismo y por tanto su coste se incluye en el coste del hardware ni de los sistemas operativos de los dispositivos Android utilizados por la misma razón.

A continuación se incluye una tabla con los costes de software del proyecto. La mayoría del software utilizado es gratuito, a excepción de Microsoft Office que es de pago y de Microsoft Visio y Microsoft Project, que aunque son de pago ha sido posible conseguir una licencia gratuita para estudiantes a través del MSDN Academic Alliance.

Costes de Software	
Concepto	Coste (€)
Eclipse IDE	0
Android Developer Tools	0
Microsoft Office Professional Plus 2010 (Word, Excel y PowerPoint)	539
Microsoft Visio Professional 2013	0
Microsoft Project Professional 2013	0
Dia Gnome 0.97.2	0
TOTAL Proyecto	539

Tabla 71. Costes de software

7.2.4. Coste total del proyecto

En esta sección se especifica el coste total del proyecto teniendo en cuenta los costes calculados en las secciones anteriores. El cálculo del coste sin IVA es la suma de los costes de personal, de hardware y de software.

A partir de este coste sin IVA calculamos el beneficio, que se ha fijado en un 50%, que se considera razonable para el proyecto desarrollado.

También a partir del coste sin IVA calculamos el margen de riesgo, que se ha fijado en un 20%, por considerarse que el riesgo no es muy elevado ya que las tecnologías utilizadas son ampliamente conocidas.

El IVA se ha calculado sobre la suma del coste sin IVA, el beneficio y el margen de riesgo y está legalmente fijado en un 21%.

El coste total del proyecto se obtiene de la suma del coste sin IVA, el beneficio, el margen de riesgo y el IVA.

Coste Total del Proyecto	
Concepto	Coste (€)
Coste de Personal	36.520,00
Coste de Hardware	121,52
Coste de Software	539,00
Coste sin IVA	37.180,52
Beneficio (50%)	18.590,26
Margen de riesgo (20%)	7.436,10
IVA (21%)	13.273,44
TOTAL Proyecto	76.480,34

Tabla 72. Coste total del proyecto

Como podemos observar en la tabla anterior y redondeando a dos decimales podemos concluir que el coste total del proyecto es de 76.480,34 €.

El presupuesto total del proyecto asciende a SETENTA Y SEIS MIL CUATROCIENTOS OCHENTA EUROS CON TREINTA Y CUATRO CÉNTIMOS.

8. CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo se presentan las conclusiones extraídas del desarrollo de este proyecto, así como las posibles líneas futuras de desarrollo del mismo.

8.1.Conclusiones

A la finalización de este proyecto cabe destacar que se han cumplido todos los objetivos del mismo que se habían especificado al comienzo del desarrollo, por lo que se considera que ha sido un éxito. Se ha desarrollado un sistema que cumple con la funcionalidad especificada y cuyo funcionamiento es correcto. El sistema desarrollado no es un prototipo ni una maqueta, sino que se trata de un sistema completo que es funcional y que cumple las expectativas esperadas del mismo. Si se deseara se podría desplegar en cualquier empresa o se podría distribuir el cliente a través de Google Play, la plataforma de distribución de aplicaciones de Google.

Este proyecto ha supuesto el reto de desarrollar un sistema bastante complejo por una única persona. El autor de este proyecto ha tenido que desempeñar todos los roles necesarios para su desarrollo y realizar todas las tareas, lo cual no es algo trivial, ya que habitualmente para un desarrollo de similar entidad se suele contar con un equipo formado por diferentes profesionales, cada uno especializado en un área concreta. Este hecho ha supuesto que el proyecto haya sido sumamente enriquecedor, ya que ha supuesto una puesta en práctica de conocimientos muy variados y a varios niveles. El autor ya poseía algunos de estos conocimientos, pero muchos otros han tenido que ser adquiridos mediante la investigación durante el desarrollo del proyecto. Por todo ello se ha conseguido una visión muy completa de lo que supone el desarrollo de un sistema de información, desde el análisis o el diseño más abstracto y general hasta la implementación de los detalles más específicos.

En cuanto a la tecnología Android, cabe destacar que el conocimiento adquirido sobre la misma en el desarrollo de este proyecto sin duda resultará muy útil ya que se trata de una tecnología muy extendida y que además se encuentra en auge. Cada vez más usuarios cuentan con dispositivos con esta tecnología, con lo que se le augura un futuro muy brillante. Además se trata de una tecnología que posibilita la reutilización de funcionalidades entre diferentes desarrolladores, lo que reduce enormemente el esfuerzo de desarrollo, al poder apoyarse en aplicaciones existentes sin necesidad siquiera de acceder a su código. En este proyecto se ha hecho uso de esta capacidad para las funcionalidades de importar ficheros, con lo que se ha evitado el trabajo de desarrollo de un gestor de ficheros, utilizándose alguno que ya esté instalado en el

dispositivo. Se ha hecho lo mismo para la funcionalidad de abrir y editar los ficheros, de forma que se utilizan aplicaciones que estén instaladas en el dispositivo para ello.

Uno de los puntos a destacar en el desarrollo de este proyecto ha sido el diseño y la implementación de un sistema que posee una arquitectura bastante compleja en cuanto a que no se trata de una arquitectura clásica, sino que toma elementos de arquitecturas diferentes. Esto ha complicado bastante el diseño y sobre todo la implementación al tener que desarrollar un sistema bastante complejo, sobre todo en el cliente.

8.2.Líneas futuras

En esta sección se exponen algunas de las posibles ampliaciones o actualizaciones de este proyecto:

1. Conexión segura entre cliente y el servidor: La conexión entre cliente y servidor podría implementarse usando un protocolo seguro como puede ser SSL o TLS de forma que se pueda tanto asegurar la identidad del servidor por parte del cliente (de forma que el cliente está seguro de que ha conectado con un servidor válido y no se trata de un servidor falso), como que se asegure que la comunicación entre ambos es confidencial de forma que cualquier atacante que pudiera estar espiando las comunicaciones no podría obtener ningún dato.
2. Posibilidad de compartir ficheros entre diferentes usuarios: Se podría implementar la posibilidad de compartir ficheros entre varios usuarios existentes en el sistema, lo que facilitaría la colaboración entre los mismos. Se podría también ofrecer la posibilidad de compartir todos los ficheros con una determinada etiqueta, de forma análoga al hecho de compartir un directorio, de forma que los usuarios no tengan que compartir los ficheros uno por uno.
3. Historial de actividad: Se podría implementar un sistema de historial de actividad que permita a los usuarios consultar cualquier actividad que se haya producido en sus cuentas, es decir, las creaciones, modificaciones y eliminaciones de ficheros, así como las modificaciones en las etiquetas de los mismos, incluyendo la fecha en que éstos fueron realizados.
4. Historial de versiones: Se podría implementar un sistema de historial de versiones de ficheros que permita recuperar las versiones antiguas de los ficheros en el sistema, de forma que se pueda deshacer cualquier cambio efectuado en el sistema. Este sistema vendría a complementar al historial de actividad propuesto en el punto anterior.
5. Compresión diferencial de los datos: Se podría implementar un sistema que cuando se modifique un archivo permita enviar únicamente la información que ha sido modificada en lugar de tener que enviar todo el fichero completo. Este tipo de técnicas se conocen como "*delta encoding*" y permiten ahorrar una cantidad considerable de tráfico de red ya que para recuperar la nueva versión del fichero sólo es necesario aplicar la información enviada a la versión original del mismo. Con este sistema se transmiten únicamente las diferencias entre las versiones del archivo. Además, este sistema permitiría ahorrar gran cantidad capacidad de almacenamiento en caso de que se implementara un sistema de control de versiones de ficheros como el descrito en el punto anterior, ya que no sería necesario almacenar todas las versiones del fichero completas, sino que se

almacenaría una versión base del mismo y las distintas actualizaciones de cada versión.

6. Almacenamiento de la información en base de datos No-SQL: Se podría cambiar la forma de almacenar los metadatos utilizados en el sistema para guardarlos en una base de datos en lugar de en ficheros como se hace en el sistema actualmente. Esta modificación se puede hacer tanto en el cliente como en el servidor como en ambos. Lo más interesante sería implementarlo en el servidor, ya que de esta manera se podría aumentar el número de usuarios conectados simultáneamente, puesto que los metadatos no tendrían que estar en memoria como actualmente. La desventaja sería una disminución de la velocidad de respuesta. El almacenamiento de los ficheros de usuario en base de datos no presenta ninguna ventaja.
7. Almacenamiento seguro de datos: Se podría implementar un sistema de almacenamiento de datos seguro para evitar que se pueda acceder a la información de forma ilícita. Para ello se podría utilizar un cifrado de datos que permita mantener la confidencialidad de los mismos y un sistema de hash seguro o firma digital que permita asegurar la integridad de los mismos en caso de que se consiga acceder a ellos de forma ilícita.
8. Funcionamiento del cliente en otras plataformas: Se podrían implementar clientes para otras plataformas sin necesidad de modificar el servidor. Cualquier cliente, independientemente de la tecnología o la plataforma del mismo, podría conectarse al servidor mientras respete el protocolo de comunicación desarrollado. Además, podría reutilizarse gran parte del código del cliente implementado en este proyecto si se opta por desarrollar el nuevo cliente en Java. Por otra parte, el servidor, al estar implementado también en Java puede funcionar en cualquier plataforma que tenga una máquina virtual de Java, con lo que puede considerarse que es muy portable a diferentes plataformas.
9. Sistema de búsqueda de ficheros: Se podría implementar un sistema de búsqueda por nombre de fichero o por contenido del mismo, que complemente a la navegación por etiquetas y haga más sencilla la navegación entre los ficheros presentes en el sistema.

9. BIBLIOGRAFÍA

- [1] Android Developer (Mayo 2013)
<http://developer.android.com>
- [2] Java (Mayo 2013)
<http://www.oracle.com/technetwork/java/index.html>
- [3] Eclipse (Mayo 2013)
<http://www.eclipse.org/>
- [4] XML (Mayo 2013)
<http://www.w3.org/TR/xml11/>
- [5] Open Handset Alliance (Mayo 2013)
<http://www.openhandsetalliance.com/>
- [6] Apache License (Mayo 2013)
<http://www.apache.org/licenses/>
- [7] Google Play (Mayo 2013)
<https://play.google.com>
- [8] Android Developer Tools (ADT) (Julio 2013)
<http://developer.android.com/tools/index.html>
- [9] W3C XML 1.1 Specification (Mayo 2013)
<http://www.w3.org/TR/xml11/>
- [10] Dropbox (Mayo 2013)
<https://www.dropbox.com>
- [11] Google Drive Support (Mayo 2013)
<https://support.google.com/drive/?hl=es>
- [12] Ubuntu One (Mayo 2013)
<https://one.ubuntu.com/services/>
- [13] Microsoft SkyDrive (Mayo 2013)
<http://windows.microsoft.com/es-es/skydrive/download>
- [14] UML Use Case Diagram Tutorial (Noviembre 2013)
http://www.sparxsystems.com/resources/uml2_tutorial/uml2_usecasediagram.html

- [15] UML Class Diagram Tutorial (Noviembre 2013)
http://www.sparxsystems.com/resources/uml2_tutorial/uml2_classdiagram.html
- [16] UML Sequence Diagram Tutorial, Sparx Systems (Noviembre 2013)
http://www.sparxsystems.com/resources/uml2_tutorial/uml2_sequencediagram.html
- [17] UML Sequence Diagram Tutorial, uml-diagrams.org (Noviembre 2013)
<http://www.uml-diagrams.org/sequence-diagrams.html>
- [18] SOAP (Mayo 2013)
<http://www.w3.org/TR/soap12-part1/>
- [19] Web Services (Mayo 2013)
<http://www.w3.org/TR/ws-arch/>
- [20] REST (Mayo 2013)
http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [21] HTTP (Mayo 2013)
<http://tools.ietf.org/html/rfc2616>

ANEXO I: MANUAL DE USUARIO

En este anexo se incluye el manual de usuario de la aplicación cliente que recibe el nombre de Synchrobox. En él se explica cómo realizar todas las funcionalidades de la aplicación.

Para facilitar la comprensión de las explicaciones se incluyen capturas de pantalla de la aplicación funcionando en un *smartphone* con sistema operativo Android. Todo el texto de la interfaz gráfica de la aplicación está escrito en lengua inglesa, ya que se ha considerado que de esta forma la aplicación podría tener más éxito en el mercado internacional. No obstante, no resultaría complicado traducir los textos de la interfaz al castellano o a cualquier otro idioma que se considere necesario.

I. 1. Registro de usuario e inicio de sesión

En primer lugar es necesario abrir la aplicación. Para ello es necesario ir al menú de aplicaciones de Android y pulsar en el icono de Synchrobox. Al hacerlo aparecerá la pantalla de bienvenida permitiendo al usuario elegir entre registro de usuario e inicio de sesión. Esta pantalla sólo aparecerá si no se ha iniciado sesión previamente. Si es así se mostrará directamente la pantalla principal de la aplicación.



Ilustración 47. Menú de aplicaciones



Ilustración 48. Pantalla de bienvenida

En el caso de pulsar el botón de registro de usuario, aparecerá la pantalla de registro en la que habrá que introducir las credenciales, es decir, nombre de usuario y contraseña de la cuenta que se quiere crear. La contraseña debe ser introducida dos veces para asegurarse de que no hay errores.

En el caso de pulsar el botón de inicio de sesión, aparecerá la pantalla correspondiente en la que será necesario introducir también las credenciales del usuario.

The screenshot shows the 'Sign Up' screen of the Synchrobox app. At the top, there is a status bar with a Wi-Fi icon, a signal strength indicator, 58% battery, and the time 02:43. Below the status bar is a dark header with the text 'Synchrobox'. The main title 'Sign Up' is centered. Below the title, there are three input fields: 'User name', 'Password', and 'Retype password'. The 'User name' field is highlighted with an orange border. At the bottom, there are two buttons: 'Sign Up' and 'Cancel'.

Ilustración 49. Pantalla de registro

The screenshot shows the 'Login' screen of the Synchrobox app. At the top, there is a status bar with a Wi-Fi icon, a signal strength indicator, 58% battery, and the time 02:43. Below the status bar is a dark header with the text 'Synchrobox'. The main title 'Login' is centered. Below the title, there are two input fields: 'User name' and 'Password'. The 'User name' field is highlighted with an orange border. At the bottom, there are two buttons: 'Login' and 'Cancel'.

Ilustración 50. Pantalla de inicio de sesión

I. 2. Eliminación de cuenta de usuario y edición de ajustes

Para eliminar una cuenta de usuario es necesario abrir el menú de opciones de la pantalla de bienvenida y pulsar en la opción de eliminar cuenta. Aparecerá una pantalla donde habrá que introducir las credenciales del usuario para poder proceder a la eliminación. Toda la información del usuario así como todos sus ficheros son borrados del sistema definitivamente por lo que esta opción debe ser utilizada con cautela.

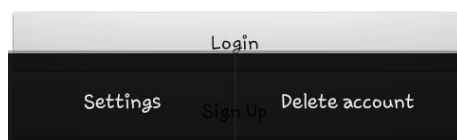


Ilustración 51. Menú de opciones de pantalla de bienvenida

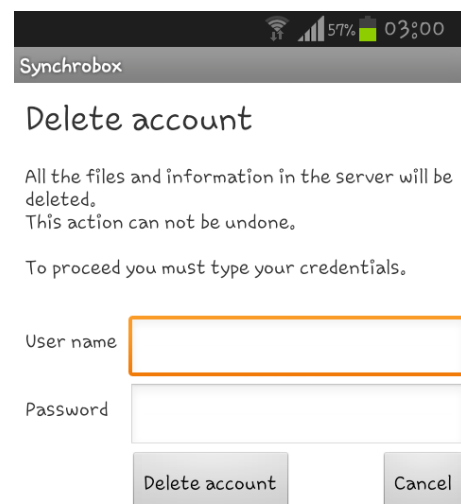


Ilustración 52. Pantalla de eliminación de cuenta de usuario

Para editar los ajustes se selecciona dicha opción también en el menú de opciones de la pantalla de bienvenida.

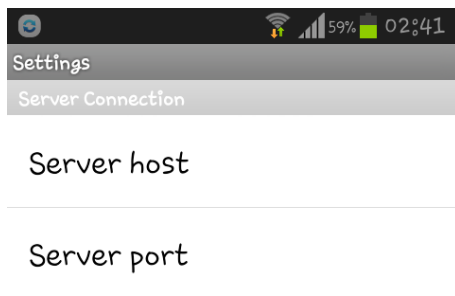


Ilustración 53. Pantalla de ajustes

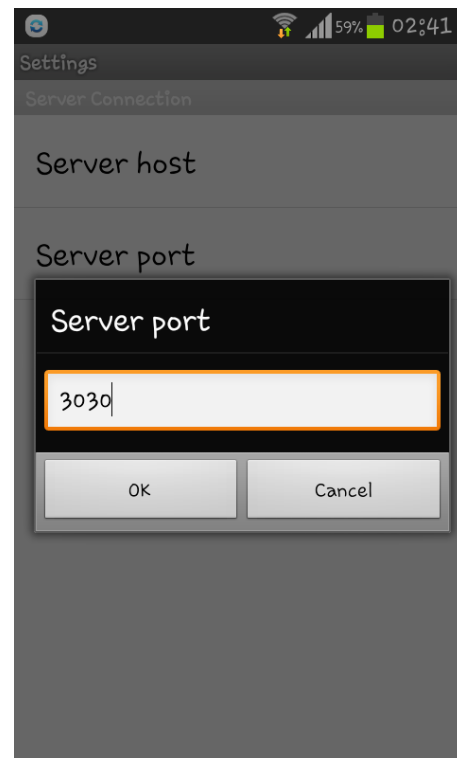


Ilustración 54. Diálogo de ajuste del puerto del servidor

Además, también se puede acceder a los ajustes desde el menú de opciones de la pantalla principal de la aplicación, aunque normalmente estos ajustes sólo se suelen editar una vez después de instalar la aplicación cliente ya que son únicamente relativos a la dirección IP y el puerto del servidor.

I. 3. Pantalla principal de la aplicación

Como se ha explicado anteriormente, una vez que el usuario ha iniciado sesión en el dispositivo se mostrará la pantalla principal de la aplicación cada vez que ésta se abra. La pantalla principal es donde se muestra la lista de ficheros así como las etiquetas. En la parte superior se muestran las etiquetas del filtro con un fondo de color azul oscuro, en caso de haber algún filtro activado. Debajo de éstas y con fondo azul claro y un icono que representa una etiqueta se muestran las etiquetas que pueden aplicarse al filtro en ese punto, es decir, las etiquetas que poseen los ficheros que se muestran en la lista excluyendo las que ya están en el filtro. Todo el funcionamiento de las etiquetas y el filtrado se explicará con detalle en las secciones correspondientes de este manual.

Por último, debajo de las etiquetas se muestran los ficheros. Para cada fichero se muestra la siguiente información:

- Icono representativo del tipo de fichero del que se trata (vídeo, audio, texto, PDF, etc.)
- Nombre del fichero incluyendo la extensión
- Estado de sincronización del fichero
- Tamaño del fichero

El estado de sincronización del fichero puede tomar 6 valores diferentes:

- **OK:** el fichero no se encuentra físicamente en el dispositivo cliente (los ficheros sólo se descargan a petición del usuario).
- **SYNCHRONIZED:** el fichero está presente físicamente en el dispositivo y su contenido se encuentra sincronizado, es decir, el cliente no conoce que haya otra nueva versión del mismo.
- **OBSOLETE:** el fichero está presente físicamente en el dispositivo pero se trata de una versión obsoleta, es decir, el cliente conoce que existe una nueva versión del fichero pero el contenido del mismo en el cliente no está actualizado.
- **NEW:** indica que el fichero es nuevo y no se ha sincronizado aún en el servidor, es decir, aún no existe en el servidor.

- **UPDATED:** indica que el fichero ha sido modificado localmente en el cliente pero los cambios aún no han sido sincronizados en el servidor.
- **CONFLICTING:** indica que existe un conflicto de versiones con el fichero. Al igual que **UPDATED** indica que el fichero ha sido modificado localmente en el cliente. El conflicto puede deberse a dos causas. La primera posible causa es que se ha modificado localmente una versión del fichero que el cliente ya conocía que estaba obsoleta (fichero con estado **OBSOLETE**). La otra posible causa es que se haya modificado la versión más reciente del fichero, pero otro cliente haya modificado esta misma versión y haya sincronizado las modificaciones en el servidor antes que este cliente. En cualquier caso, el conflicto de versiones se reduce al hecho de que se produzca una modificación de versiones no lineal o no incremental en el tiempo por diferentes clientes.

Además, la pantalla principal cuenta con un menú de opciones que permite realizar algunas funcionalidades que se explicarán más adelante.

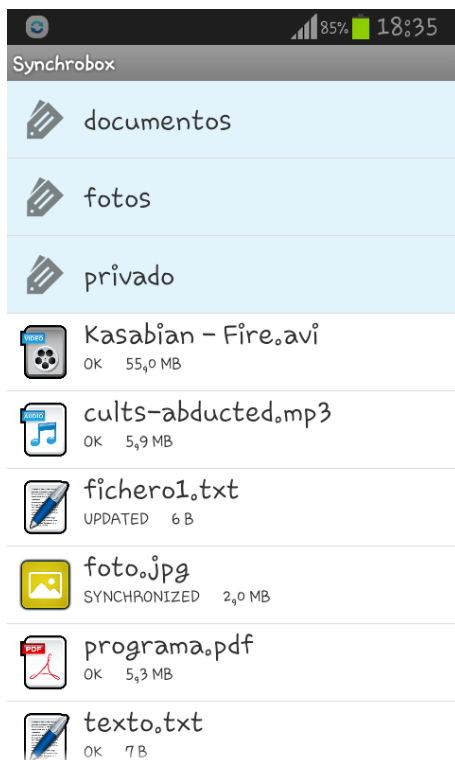


Ilustración 55. Pantalla principal

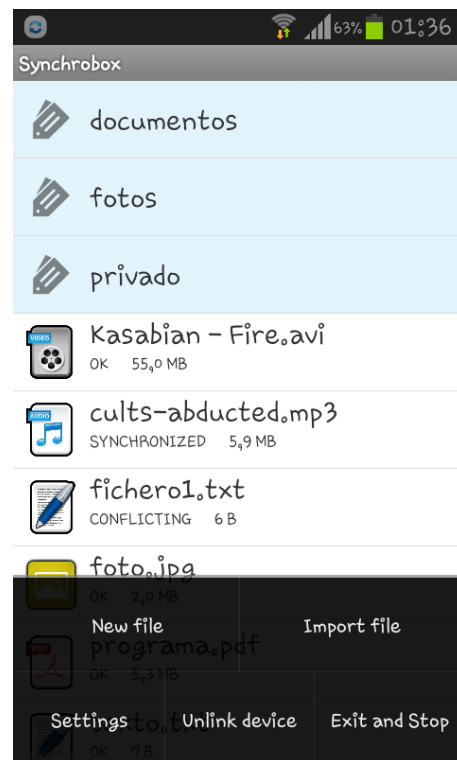


Ilustración 56. Menú de opciones de la pantalla principal

Una vez que se ha iniciado sesión de usuario, se muestra el icono de Synchrobox en la zona de notificaciones de la barra de estado del dispositivo (parte superior izquierda de la pantalla). Este icono permanece activo continuamente aunque se cierre la pantalla de la aplicación o se cambie a otra aplicación diferente. Nos indica que el servicio de la aplicación está iniciado y que por tanto la aplicación está funcionando en segundo plano aunque no se muestre su interfaz gráfica. Si se abre la bandeja de notificaciones deslizando hacia abajo desde la barra de estado se muestra que la aplicación está en curso. Si se pulsa sobre la notificación la aplicación se abre y muestra la pantalla principal (es una forma alternativa de abrir la aplicación).

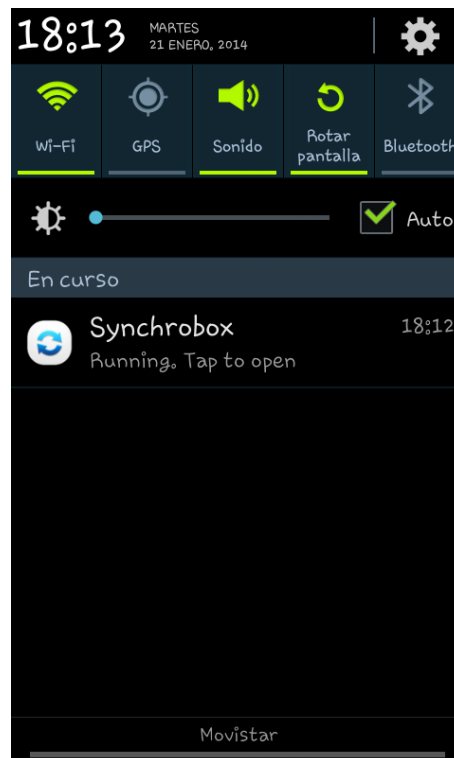


Ilustración 57. Bandeja de notificaciones

I. 4. Apertura y edición de ficheros

Para abrir un fichero sólo hay que pulsar sobre él en la lista de ficheros de la pantalla principal de la aplicación. Si el fichero necesita ser descargado del servidor porque no se encuentre en el dispositivo o porque esté obsoleto, se muestra un diálogo que indica que se está descargando y que permite cancelar la descarga si se desea.

El fichero se abrirá en una aplicación externa que sirva para abrir ese tipo de fichero. En caso de haber más de una aplicación que pueda abrir ese tipo de fichero instalada en el dispositivo, se mostrará una lista con ellas para que el usuario elija una. En caso de no existir ninguna aplicación que pueda abrir el fichero se mostrará un mensaje indicándolo.

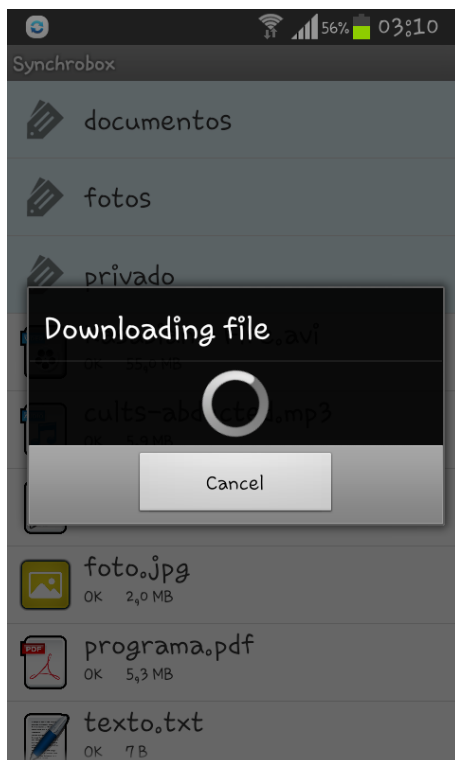


Ilustración 58. Diálogo de descarga de fichero

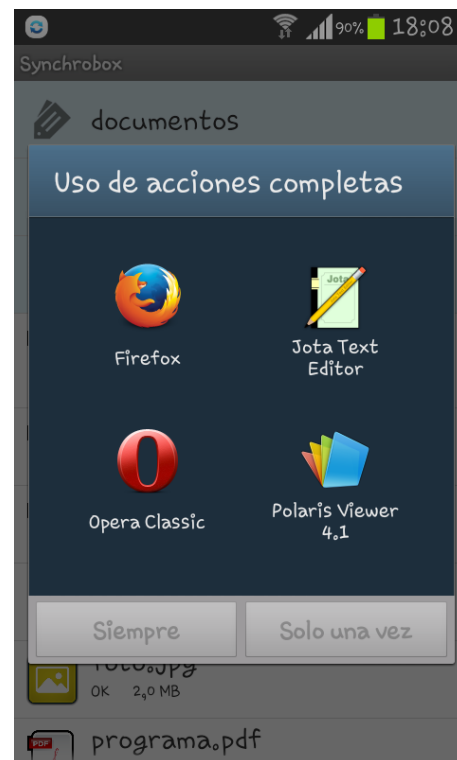


Ilustración 59. Pantalla de elección de aplicación de apertura de fichero

Si el fichero que se desea abrir está obsoleto (su estado es `OBSOLETE` como ya se ha explicado anteriormente) y además no se puede conectar al servidor para descargar la versión más reciente del mismo, se muestra un diálogo indicándolo y advirtiéndole de que si se modifica el fichero puede ocurrir un conflicto de versiones del fichero. Se permite al usuario decidir si quiere intentar descargarlo, abrir la versión obsoleta o cancelar la operación.

Si existe un conflicto de versiones con el fichero que se va a abrir, es decir, el estado del fichero es `CONFLICTING`, entonces al pulsar en el fichero aparecerá una pantalla indicando que hay un conflicto de versiones. Se muestra la fecha de sincronización de la versión que hay en el dispositivo y de la versión del servidor y se permite al usuario decidir con cuál de estas dos versiones quiere quedarse.

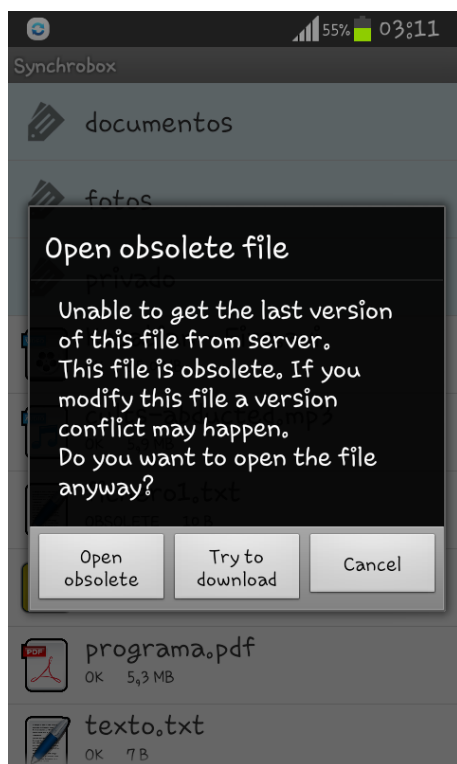


Ilustración 60. Diálogo de apertura de fichero obsoleto

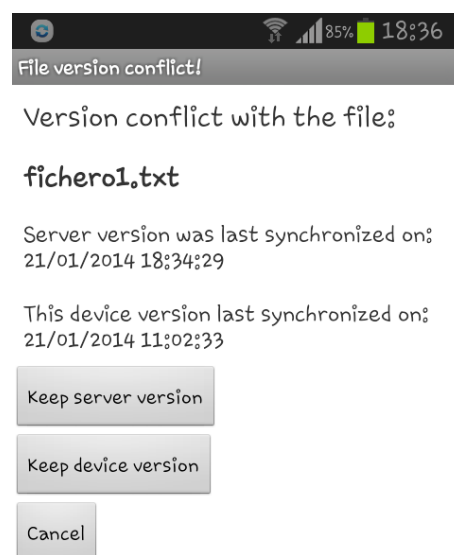


Ilustración 61. Pantalla de conflicto de versiones de fichero

Para editar un fichero éste debe abrirse como se ha explicado antes y se editará con la aplicación con la que se haya abierto, si es que ésta permite la edición. En el momento que se guarden los cambios realizados en el fichero por dicha aplicación externa, el cliente detectará el cambio e intentará sincronizarlo en el servidor.

I. 5. Creación de ficheros

Para crear un nuevo fichero hay que abrir el menú de opciones de la pantalla principal y seleccionar la opción de crear nuevo fichero. Aparecerá un cuadro de diálogo que permitirá introducir el nombre del nuevo fichero. Si se introduce un nombre no válido o que ya exista en el sistema se mostrará un mensaje indicando el error.

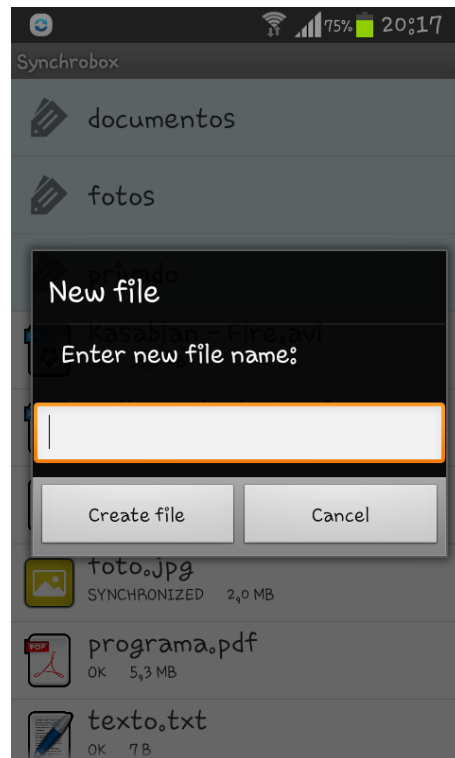


Ilustración 62. Diálogo de creación de fichero

I. 6. Importación de ficheros

Para importar un fichero se selecciona la opción de importar en el menú de opciones de la pantalla principal. En ese momento se abrirá una aplicación externa que permita seleccionar el fichero a importar, típicamente se tratará de un gestor de ficheros. En caso de haber más de una aplicación que permita seleccionar el fichero instalada en el dispositivo, se mostrará una lista con ellas para que el usuario elija una. En caso de no existir ninguna aplicación que permita seleccionar el fichero se mostrará un mensaje indicándolo.

Una vez que se haya seleccionado el fichero en la aplicación externa aparecerá un diálogo que permite introducir el nombre con el que el fichero será importado. Por defecto aparecerá el nombre original del fichero permitiendo al usuario utilizarlo tal cual o modificarlo si lo desea. Al igual que ocurre al crear un nuevo fichero si se introduce un nombre no válido o que ya exista en el sistema se mostrará un mensaje indicando el error.

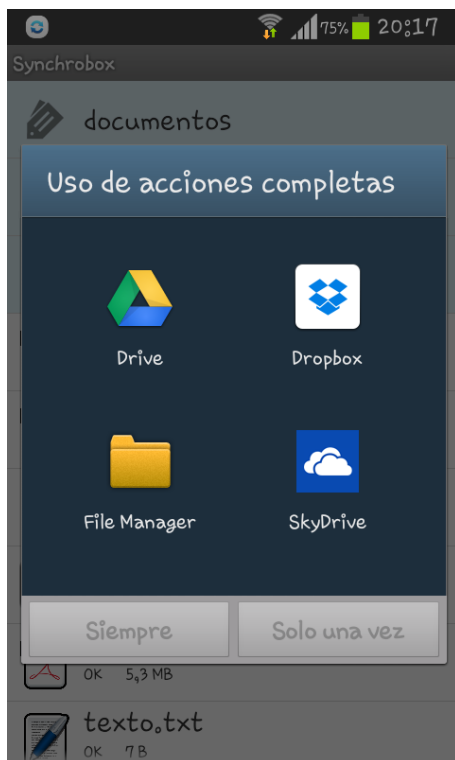


Ilustración 63. Diálogo de elección de aplicación para seleccionar fichero a importar

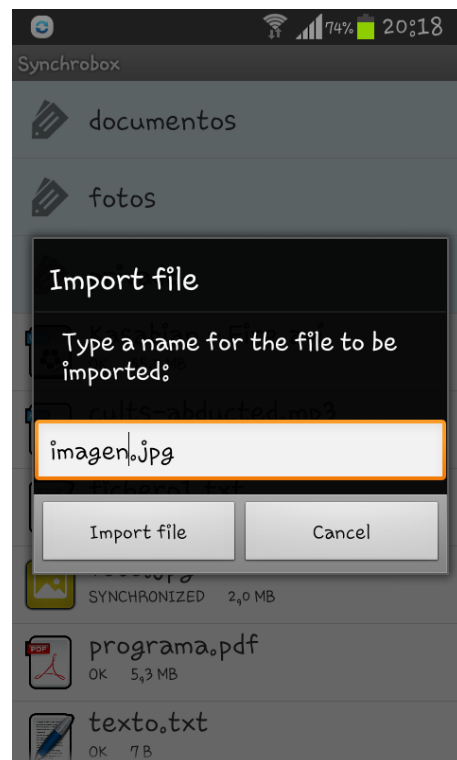


Ilustración 64. Diálogo de importación de fichero

I. 7. Borrado de ficheros

Para borrar un fichero se debe abrir el menú contextual del fichero. Para ello se debe mantener pulsado el fichero en la lista de ficheros de la pantalla principal. En el menú contextual se seleccionará la opción de borrar y el fichero será eliminado.

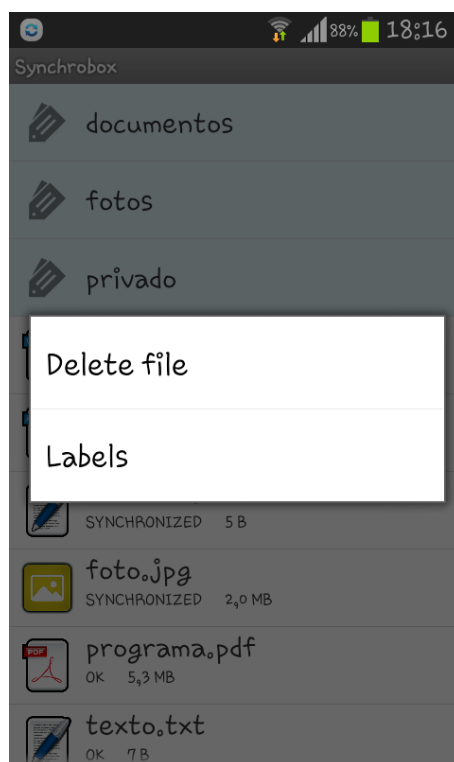


Ilustración 65. Menú contextual de fichero

I. 8. Visualización y adición de etiquetas

Para visualizar las etiquetas que tiene un fichero se debe abrir el menú contextual del fichero (de la misma forma que se abre para el borrado del fichero). En el menú contextual se seleccionará la opción de etiquetas. A continuación se mostrará una pantalla en la que aparece el nombre del fichero y una lista con todas las etiquetas que tiene. Para volver a la pantalla principal se pulsará el botón de retroceso.

Si se desea añadir una etiqueta es necesario estar en la pantalla de etiquetas del fichero que se ha explicado en el párrafo anterior. Se pulsará el botón de añadir etiqueta que se encuentra en la zona superior derecha o se mostrará el menú de opciones y se seleccionará la de añadir etiqueta, cualquiera de las dos opciones está disponible. Entonces se mostrará un diálogo que permite introducir la etiqueta que se desea añadir. Si se introduce un nombre de etiqueta no válido o que ya tenga ese fichero se mostrará un mensaje indicando el error. La adición de la etiqueta o etiquetas al fichero se hará realmente efectiva cuando se vuelva a la pantalla principal pulsado el botón de retroceso.

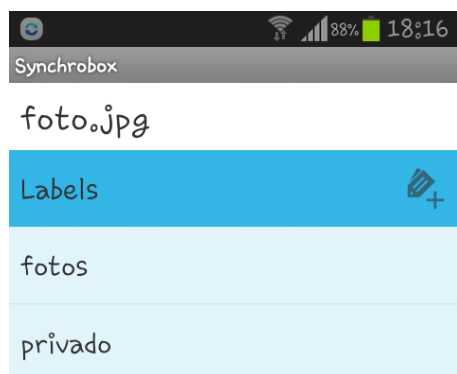


Ilustración 66. Pantalla de etiquetas

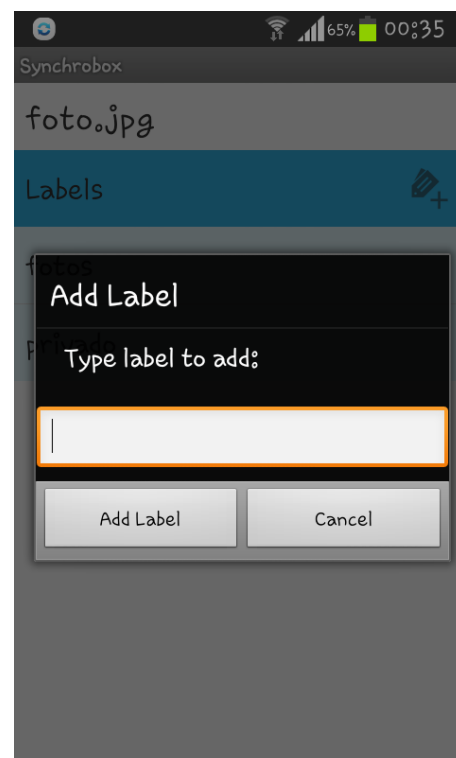


Ilustración 67. Diálogo de adición de etiqueta

I. 9. Eliminación de etiquetas

Para eliminar una etiqueta, igual que para añadirla, es necesario estar en la pantalla de etiquetas del fichero. Una vez en la pantalla de etiquetas se debe abrir el menú contextual de la etiqueta que se desea eliminar. Para ello se debe mantener pulsada la etiqueta en la lista de etiquetas del fichero. En el menú contextual se seleccionará la opción de borrar etiqueta.

Igual que ocurriría con la adición de etiquetas, la eliminación de la etiqueta o etiquetas del fichero se hará realmente efectiva cuando se vuelva a la pantalla principal pulsado el botón de retroceso. Cabe destacar que se pueden realizar cuantos cambios en las etiquetas se desee, tanto adiciones como eliminaciones, que todos ellos se harán efectivos al pulsar el botón de retroceso.

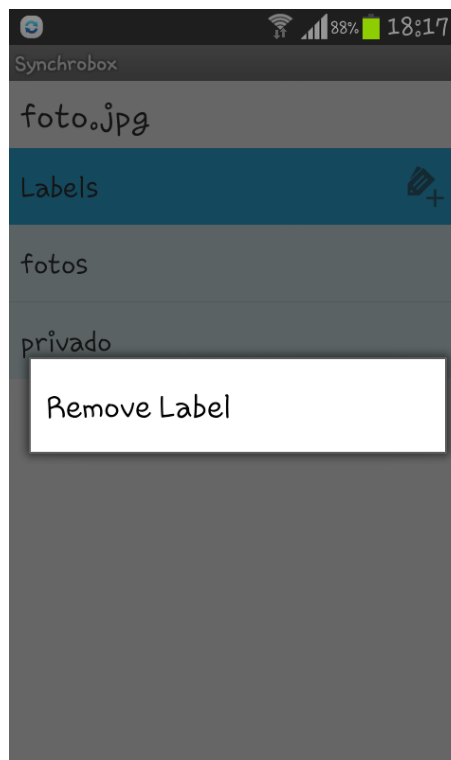


Ilustración 68. Menú contextual de etiqueta

I. 10. Filtrado de ficheros por etiquetas

Para filtrar los ficheros por una determinada etiqueta de forma que sólo se muestren los ficheros que tienen dicha etiqueta es necesario encontrarse en la pantalla principal de la aplicación. Se debe pulsar la etiqueta por la cual se desea filtrar. Las etiquetas por las que es posible realizar el filtrado se muestran con fondo azul claro encima de la lista de ficheros y precedidas por un icono que representa una etiqueta.

En el momento en que se pulse una etiqueta para realizar el filtrado ésta pasará a formar parte del filtro y se mostrará en la parte superior de la pantalla con fondo azul oscuro. A partir de este momento se muestran únicamente los ficheros que tienen dicha etiqueta en la lista de ficheros. Además, se muestra el resto de etiquetas que poseen dichos ficheros, es decir, se muestran en la lista de etiquetas las que poseen los ficheros que se encuentran en ese momento en la lista de ficheros exceptuando las que están en el filtro. Esto permite seguir filtrando por alguna de estas etiquetas pulsando sobre ella, es decir, se puede añadir alguna de estas etiquetas al filtro con lo que se consigue refinar la búsqueda de ficheros.

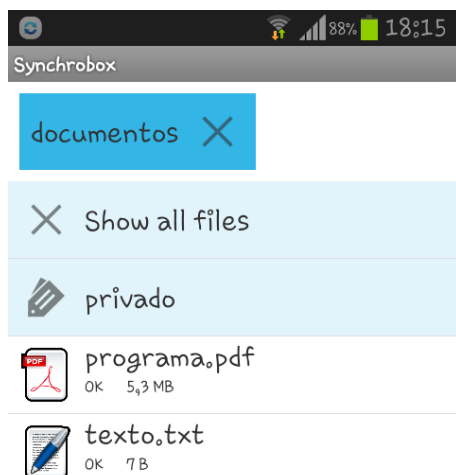


Ilustración 69. Filtrado por etiqueta “documentos”

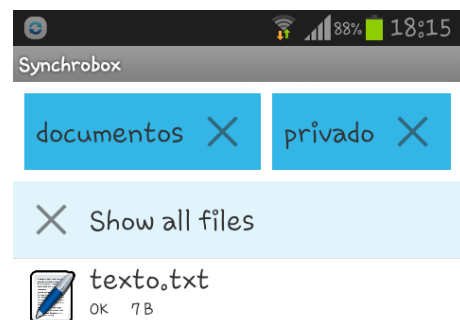


Ilustración 70. Filtrado por etiquetas “documentos” y “privado”

I. 11. Eliminación de etiqueta del filtro y desactivación del filtrado por etiqueta

Para eliminar una etiqueta del filtro, desde la pantalla principal, se pulsa en el icono en forma de X que se encuentra a la derecha del nombre de la etiqueta en el filtro. Con esto se consigue que se muestren también los ficheros que no poseen dicha etiqueta pero que poseen el resto de etiquetas del filtro, en caso de que las haya. Es posible añadir o eliminar etiquetas al filtro en cualquier orden o las veces que se desee.

Además, se puede desactivar el filtrado por etiquetas pulsando en el elemento que indica “mostrar todos los ficheros” que aparece el primero en la lista de etiquetas cuando hay algún filtro activo. Este elemento desaparece cuando no se está filtrando por ninguna etiqueta. Realmente esta operación es un atajo que elimina todas las etiquetas del filtro para no tener que eliminarlas una por una.

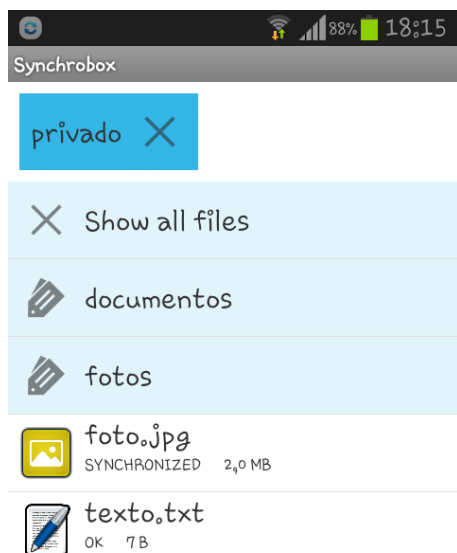


Ilustración 71. Filtrado por etiqueta "privado"

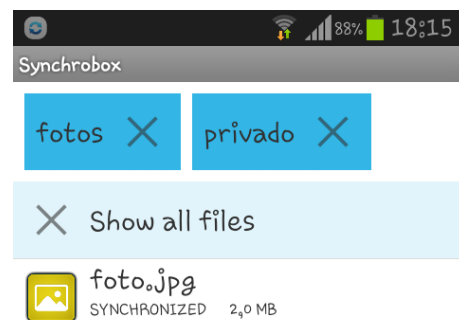


Ilustración 72. Filtrado por etiquetas "fotos" y "privado"

I. 12. Cierre de la aplicación y cierre de sesión de usuario

Como ya se ha explicado anteriormente, una vez que se ha iniciado sesión de usuario, se muestra el icono de Synchrobox en la zona de notificaciones de la barra de estado del dispositivo permanentemente aunque se cierre la pantalla de la aplicación o se cambie a otra aplicación diferente. Esto ocurre así para indicar que el servicio de la aplicación está iniciado y por tanto la aplicación está funcionando en segundo plano aunque no se muestre su interfaz gráfica.

Si se desea cerrar completamente la aplicación se debe abrir el menú de opciones de la pantalla principal y seleccionar la opción de salir y parar la aplicación. Esto detiene su servicio en segundo plano además de cerrar la interfaz gráfica y cerrar las conexiones abiertas, con lo que la aplicación no consumirá ningún recurso del dispositivo, aunque evidentemente la sincronización se desactivará hasta que se vuelva a abrir la aplicación de nuevo. El icono de la aplicación desaparece de la zona de notificaciones de la barra de estado del dispositivo.

Si lo que se desea es cerrar la sesión de usuario en ese dispositivo, es decir, desvincular el dispositivo de la cuenta del usuario se debe seleccionar la opción correspondiente en el menú de opciones de la pantalla principal. Aparecerá un diálogo que advertirá al usuario y le solicitará que confirme la acción. En caso de confirmar el cierre de sesión se elimina toda la información del usuario y sus ficheros del dispositivo, aunque todo ello permanecerá en el servidor.

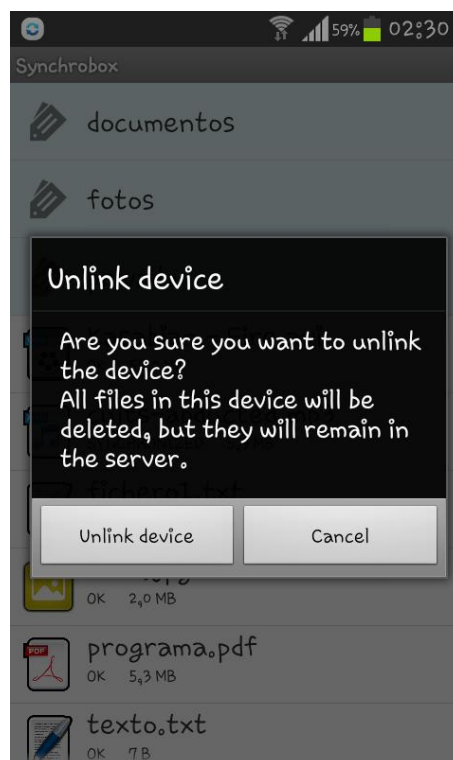


Ilustración 73. Diálogo de cierre de sesión